M 2018

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MODELLING CRACK PROPAGATION USING THE FINITE ELEMENT METHOD AND RADIAL POINT INTERPOLATION MESHLESS METHODS

LUÍS DANIEL COSTA RAMALHO

DISSERTAÇÃO DE MESTRADO APRESENTADA
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM
MECÂNICA COMPUTACIONAL

Faculdade de Engenharia da Universidade do Porto

Modelling Crack Propagation Using the Finite Element Method and Radial Point Interpolation Meshless Methods

Luís Daniel Costa Ramalho

Dissertação realizada no âmbito do
Mestrado em Mecânica Computacional

Orientador: Prof. Dr. Jorge Américo De Oliveira Pinto Belinha
Co-orientador: Prof. Dr. Renato Manuel Natal Jorge

Junho de 2018

Resumo

A previsão da propagação de fendas é um problema de engenharia importante. Nesta tese, foi desenvolvido um algoritmo para a previsão de propagação de fenda, este algoritmo funciona com três métodos numéricos diferentes, o método dos elementos finitos, o *radial point interpolation method* e o *natural neighbour radial point interpolation method*.

Este algoritmo propaga a fenda de forma iterativa até que um dado comprimento máximo de fenda, previamente estabelecido, seja alcançado. A direcção de propagação de fenda é calculada usando o critério da tensão tangencial máxima. Para este cálculo é usada uma média ponderada das tensões dos nós à volta da fenda, de modo a suavizar o campo de tensões na vizinhança da fenda. Este algoritmo também assegura que há sempre oito elementos e nove nós à volta da ponta da fenda, a não ser que esta esteja muito perto de uma fronteira, nesse caso podem haver menos nós e elementos à volta da fenda.

Para verificar a validez do algoritmo, foram realizados diversos exemplos numéricos de referência, alguns deles com várias pontas de fenda. A análise dos referidos exemplos permitiu mostrar que este algoritmo obtém resultados precisos quando comparados com os resultados encontrados em trabalhos anteriores, desde que a malha inicial não seja demasiado grosseira. Este algoritmo também mantem as malhas bastante regulares durante o processo de propagação, com poucos elementos distorcidos, o que geralmente é um dos maiores problemas quando se efectua a propagação de fenda com o método dos elementos finitos.

Outra conclusão deste trabalho é que com este algoritmo as diferenças entre os caminhos de propagação de fenda calculados com os três métodos distintos são geralmente bastante pequenas.

Palavras-chave: Propagação de fenda, FEM, métodos sem malha, NNRPIM, RPIM, mecânica da fractura.

Abstract

The prediction of crack propagation is an important engineering problem. In this work, an algorithm to predict crack propagation was developed. This algorithm works with three different numerical methods, the finite element method (FEM), the radial point interpolation method and the natural neighbour radial point interpolation method.

The proposed algorithm extends the crack iteratively until a threshold maximum crack length is achieved. The crack propagation direction is calculated using the maximum tangential stress criterion. In this calculation, a weighted average of the stresses of the nodes around the crack tip is considered to smoothen the stress field in the vicinity of the crack tip. The algorithm also ensures that there are always at least eight elements and nine nodes surrounding the crack tip, unless the crack tip is close to a domain boundary, in which case there can be fewer elements and nodes around the crack tip.

To ascertain the validity of the algorithm several benchmark tests were performed, some of them with multiple crack tips, which showed that this algorithm would lead to accurate crack paths when compared to findings from previous papers, as long as the mesh is not too coarse. This algorithm also leads to very regular meshes during the propagation process, with very few distorted elements, which is generally one of the main problems when calculating crack propagation with the finite element method.

Another finding of this work is that using this algorithm, it is possible to achieve very similar crack paths for the different methods – the FEM or meshless methods.

Keywords: Crack propagation, FEM, meshless methods, NNRPIM, RPIM, fracture mechanics.

Agradecimentos

Aos meus pais, Alice Maria e António Eduardo, que durante todos estes anos me apoiaram sem pedirem muito em troca.

Um especial agradecimento ao doutor Jorge Belinha que incutiu em mim o gosto da programação e o interesse em métodos numéricos. Queria também agradecer a sua disponibilidade para ser o orientador desta tese e pelas reuniões semanais que me ajudaram a ver os problemas que surgiram na elaboração desta tese de pontos de vista diferentes.

Ao meu irmão David pela oferta de estadia durante este ano e pelas conversas durante as viagens à terra natal, Vila Real.

À minha irmã Rita por ser uma fonte de alegria até nos momentos mais difíceis da elaboração desta tese e por ser a minha parceira em jogos de tabuleiro.

Ao resto da minha família, em particular ao meu padrinho, Jorge Ramalho, por, ao longo de tantos anos, me ter levado a tantas corridas em Portugal, o que ajudou a cimentar o meu interesse pela engenharia mecânica.

A todos os meus amigos e colegas de curso que proporcionaram momentos de descontracção, mas também momentos de reflexão.

Institutional Acknowledgments and Funding

The author truly acknowledges the work conditions provided by the Applied Mechanics Division (SMAp) of the department of mechanical engineering (DEMec) of Faculty of Engineering of the University of Porto (FEUP), and by the MIT-Portugal project "MIT-EXPL/ISF/0084/2017", funded by Massachusetts Institute of Technology (USA) and "Ministério da Ciência, Tecnologia e Ensino Superior - Fundação para a Ciência e a Tecnologia" (Portugal).

Additionally, the authors gratefully acknowledge the funding of Project NORTE-01-0145-FEDER-000022 - SciTech - Science and Technology for Competitive and Sustainable Industries, cofinanced by Programa Operacional Regional do Norte (NORTE2020), through Fundo Europeu de Desenvolvimento Regional (FEDER).

Finally, the author acknowledges the synergetic collaboration with the collaborators of "Computational Mechanics Research Laboratory CMech-Lab" (ISEP/FEUP/INEGI), and its director, Prof.Dr. Jorge Belinha, and its senior advisors, Prof.Dr. Renato Natal Jorge and Prof.Dr. Lúcia Dinis.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Failures can be a major cause of accidents and of equipment breaking, and it generally starts with a small crack that grows overtime when subjected to stress. Because of this, it is important to study and predict the behaviour of crack growth.

This thesis covers the use of meshless methods and the finite element method (FEM) to solve crack propagation problems with brittle materials, being the meshless methods used the Radial Point Interpolation Method (RPIM) and the Natural Neighbour Radial Point Interpolation Method (NNRPIM).

The use of different numerical methods will enable the comparison between the widely used FEM and the meshless methods used in this work, and the verification of the quality of the solutions provided by the RPIM and the NNRPIM in this type of problem, when compared to FEM solutions.

Meshless methods are a type of numerical method used to solve partial differential equations. Meshless methods do not require the use of a mesh, or grid of elements, to discretize the problems, hence their name. However, in some meshless methods a background mesh is employed to create the integration points [1]. These methods, instead of having a mesh of elements, only have a structured grid of integration points, their respective influence domains and nodes. The connectivity between nodes is achieved through the influence domains, while the FEM applies a mesh to the problem domain, which divides the domain into "smaller domains" called elements [1]. These elements are connected and interact with each other, they are generally in the shape of quadrilaterals or triangles for 2D problems and they are composed of integration points and nodes.

Because the meshless methods do not use elements, they can be more easily applied to complex geometries, as opposed to FEM [2]. Another advantage of the meshless methods when compared to the FEM is that, since they do not possess elements, the propagation of a crack is much simpler to solve as there is no need of remeshing [3]. Nonetheless, the algorithm created for this work is intended to work regardless of the method used whether it is the FEM, the RPIM or the NNRPIM, so this advantage will not be observed in this work.

This work intends to improve on the work developed by Azevedo [4], that developed a crack propagation algorithm for the NNRPIM. The improvement done here focuses mainly on

developing an algorithm that works and obtains good results using the NNRPIM, but also other numerical methods.

The crack propagation algorithm was developed using the software Matlab to be incorporated in the Matlab program FEMAS developed by Jorge Belinha.

In this thesis, the three numerical methods will be used to conduct benchmark tests and the results of these tests will be compared to results found in previous papers.

1.1. Thesis Outline

This thesis is divided in six chapters.

The first chapter is the introduction, which gives a brief overview of the goals of the thesis and a brief explanation of the differences between meshless methods and the finite element method.

The second chapter presents the solid mechanics fundamentals and the discretization of solid mechanics. Then, there is a presentation of the fundamental concepts of fracture mechanics and a state of the art about the propagation criteria found in literature.

In chapter three, the base FEM, RPIM and NNRPIM formulations are presented, these are common to any algorithm using any of these numerical methods.

Chapter four presents the previous algorithms and solutions, found in literature, that approached the numerical prediction of crack propagation, for each numerical method used.

The fifth chapter presents the developed algorithm and the sixth chapter presents several benchmark tests and the corresponding crack paths obtained.

In the seventh and final chapter the conclusions of this work are presented and discussed, and possible future improvements and expansions of this work are analysed.

Chapter 2

Solid Mechanics

2.1. Fundamentals

Solid mechanics define the stress field for a given solid subjected to boundary conditions, external forces and imposed or constrained displacements, this stress field then leads to strains.

In this work, the problems considered belong to the linear-elastic field, therefore, only the linear-elastic theory is presented here. In this case, linear refers to the material behaviour and the consequent relationship between the stresses and strains, meaning that the strains evolve linearly with the stresses. Elastic refers to the material behaviour of the deformed solid after the applied loads are removed, meaning that it returns to its original undeformed shape when the loads are removed [4].

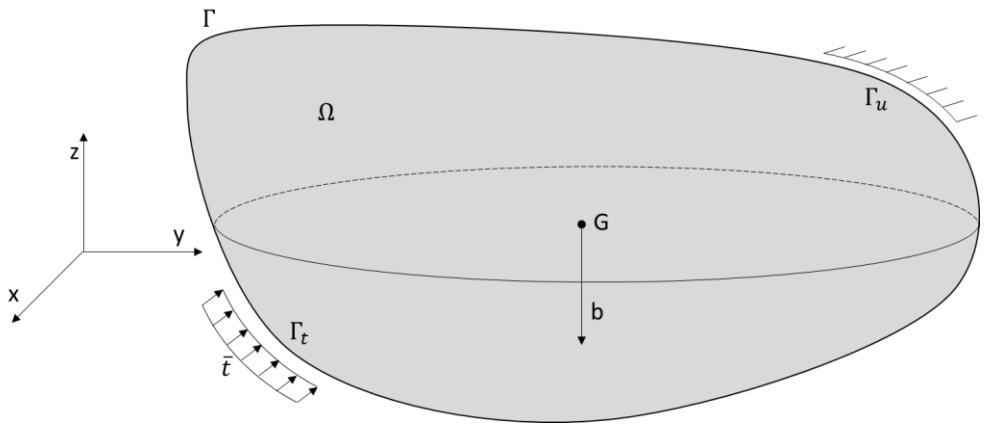

Figure 1 - Standard solid mechanics problem

Figure 1 shows the standard solid mechanics problem, where the solid is the domain Ω, subjected to a body force b and with a boundary Γ, which has a force applied in Γ_t and its movement is constrained in Γ_u. The force applied is classified as a natural boundary condition and the displacement constrains are defined as an essential boundary condition [1].

2.1.1. Stress and Strain

In solid mechanics, stress is defined by a stress tensor for a given point in the body:

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{1}$$

This is a symmetric tensor because $\sigma_{ij} = \sigma_{ji}$, therefore, to simplify, it can also be written as a vector, which is simpler to use in computational mechanics:

$$\sigma = \{\sigma_{xx} \quad \sigma_{yy} \quad \sigma_{zz} \quad \sigma_{xy} \quad \sigma_{yz} \quad \sigma_{zx}\}^{\mathrm{T}} \tag{2}$$

The strain tensor can be defined as:

$$\varepsilon = \begin{bmatrix} \varepsilon_{xx} & \frac{1}{2}\gamma_{xy} & \frac{1}{2}\gamma_{xz} \\ \frac{1}{2}\gamma_{yx} & \varepsilon_{yy} & \frac{1}{2}\gamma_{yz} \\ \frac{1}{2}\gamma_{zx} & \frac{1}{2}\gamma_{zy} & \varepsilon_{zz} \end{bmatrix} \tag{3}$$

Since it is also symmetric, it can also be simplified as follows:

$$\varepsilon = \{\varepsilon_{xx} \quad \varepsilon_{yy} \quad \varepsilon_{zz} \quad \gamma_{xy} \quad \gamma_{yz} \quad \gamma_{zx}\}^{\mathrm{T}} \tag{4}$$

2.1.2. Principal Stresses and Principal Strains

It is possible to obtain a stress vector, for any given stress matrix, that is parallel to a given vector, $\boldsymbol{n}$ [1]:

$$\boldsymbol{t}^{(n)} = \boldsymbol{n} \cdot \sigma_{\mathrm{ij}} = [n_1 \quad n_2 \quad n_3] \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{5}$$

As a second order tensor, the stress tensor has eigenvalues, called principal stresses, these give the stress magnitude in the directions where there is no shear stress, being those directions the principal directions. Using equation 5:

$$\boldsymbol{t}_i^{(n)} = \lambda \boldsymbol{n}_i \Rightarrow \sigma_{ij}\boldsymbol{n}_j = \lambda \delta_{ij}\boldsymbol{n}_j \Rightarrow \left(\sigma_{ij} - \lambda\delta_{ij}\right)\boldsymbol{n}_j = 0 \tag{6}$$

where λ is a constant of proportionality, an unknown correspondent to the magnitude of the principal stresses, and δ_{ij} is the Kronecker delta. To obtain a solution where $\boldsymbol{n}_j \neq 0$ the matrix determinant must equal to zero as follows [1]:

$$\left|\sigma_{ij} - \lambda\delta_{ij}\right| = \begin{vmatrix} \sigma_{xx} - \lambda & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} - \lambda & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} - \lambda \end{vmatrix} = 0 \tag{7}$$

which leads to the following cubic equation:

$$\left|\sigma_{ij} - \lambda\delta_{ij}\right| = -\lambda^3 + I_1\lambda^2 - I_2\lambda + I_3 = 0 \tag{8}$$

where I_1, I_2 and I_3 are the invariants of the stress matrix and they are defined as:

$$\begin{aligned} I_1 &= \sigma_{kk} \\ I_2 &= \frac{1}{2}\left(\sigma_{ii}\sigma_{jj} - \sigma_{ij}\sigma_{ji}\right) \\ I_3 &= \det(\sigma) \end{aligned} \tag{9}$$

Since equation 8 is a cubic equation, it has 3 different solutions for λ, which are the magnitude of the principal stresses σ_1, σ_2 and σ_3, being σ_1 the highest solution and σ_3 the lowest. To obtain the principal directions, equation 6 is used, substituting λ by each principal stress, being the n_j obtained the principal direction corresponding to that principal stress.

To obtain the principal strains, a procedure similar to the one described for the principal stresses is used, but substituting the stress tensor by the corresponding strain tensor. Additionally, following the same process, and considering the obtained principal strains, it is also possible to calculate their principal directions.

2.1.3. Compliance Matrix

The compliance matrix gives the relationship between stresses and strains, and it is equal to [1]:

$$\boldsymbol{s} = \begin{bmatrix} \frac{1}{E_{xx}} & -\frac{\nu_{yx}}{E_{yy}} & -\frac{\nu_{zx}}{E_{zz}} & 0 & 0 & 0 \\ -\frac{\nu_{xy}}{E_{xx}} & \frac{1}{E_{yy}} & -\frac{\nu_{zy}}{E_{zz}} & 0 & 0 & 0 \\ -\frac{\nu_{xz}}{E_{xx}} & -\frac{\nu_{yz}}{E_{yy}} & \frac{1}{E_{zz}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{zx}} \end{bmatrix} \tag{10}$$

Where E_{ii} is the Young's modulus of the material in the direction i, determined experimentally and ν_{ij} is the Poisson ratio of the material which is the ratio of deformation in the direction j when a force is applied in the direction i. If the material is isotropic the Young's modulus and Poisson ratio remain the same for all directions, simplifying this equation. G_{ij} is the shear modulus and it is defined by:

$$G_{ij} = \frac{E_{ii}}{2\left(1 + \nu_{ij}\right)} \tag{11}$$

In the 2D case, which is the object study of this work, this matrix can be simplified in two different ways. The first one is called plane stress theory which assumes that the stresses in direction z are null, $\sigma_{zx} = \sigma_{zy} = \sigma_{zz} = 0$, this leads to the following compliance matrix:

$$\boldsymbol{s} = \begin{bmatrix} \frac{1}{E_{xx}} & -\frac{\nu_{yx}}{E_{yy}} & 0 \\ -\frac{\nu_{xy}}{E_{xx}} & \frac{1}{E_{yy}} & 0 \\ 0 & 0 & \frac{1}{G_{xy}} \end{bmatrix} \tag{12}$$

The other simplification that can be done in 2D is the plane strain theory, which assumes that the strains are null in the z direction, $\varepsilon_{zx} = \varepsilon_{zy} = \varepsilon_{zz} = 0$, this results in the following compliance matrix:

$$\boldsymbol{s} = \begin{bmatrix} \frac{1}{E_{xx}} - \frac{\nu_{zx}\nu_{xz}}{E_{xx}} & -\frac{\nu_{yx}}{E_{yy}} - \frac{\nu_{zx}\nu_{yz}}{E_{yy}} & 0 \\ -\frac{\nu_{xy}}{E_{xx}} - \frac{\nu_{zy}\nu_{xz}}{E_{xx}} & \frac{1}{E_{yy}} - \frac{\nu_{zy}\nu_{yz}}{E_{y}} & 0 \\ 0 & 0 & \frac{1}{G_{xy}} \end{bmatrix} \tag{13}$$

The relationship between stresses and strains is given by Hooke's law:

$$\begin{aligned} \boldsymbol{\sigma} &= \boldsymbol{c}\,\boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} &= \boldsymbol{s}\,\boldsymbol{\sigma} \end{aligned} \tag{14}$$

where $\mathbf{c} = \boldsymbol{s}^{-1}$ which is the constitutive matrix.

2.2. Galerkin Weak Formulation

The strong formulation consists of the system of partial differential equations that govern a given problem, where solving these allows to obtain the exact solution of the problem. However, in complex engineering problems, it is difficult to find a solution with the strong formulation [1].

The weak formulation requires a weaker consistency on the used approximation, or interpolation functions. Weak formulations can produce a stable algebraic system of equations and give a discretized system of equations, this leads to more accurate results [1]. Thus, the weak formulation is preferred to obtain an approximated solution.

Consider a solid with a domain Ω and its boundary Γ, as in Figure 1, subjected to the boundary conditions and forces presented there. The Galerkin weak formulation is a variational principle based on the energy principle. Between all the solutions that satisfy the compatibility equations, the essential boundary conditions and the initial and final conditions, the real solution is the one that minimizes the Lagrangian functional L [1]:

$$L = T - U + W_f \tag{15}$$

where T is the kinetic energy, U the strain energy and W_f the work produced by the external forces. The kinetic energy is defined as [1]:

$$T = \frac{1}{2}\int_{\Omega} \rho\, \dot{\boldsymbol{u}}^T \dot{\boldsymbol{u}}\, \mathrm{d}\Omega \tag{16}$$

being $\dot{\boldsymbol{u}}$ the displacement first derivative with respect to time (the velocity) and ρ the density of the solid. The strain energy is defined as [1]:

$$U = \frac{1}{2}\int_{\Omega} \varepsilon^T \sigma \, d\Omega \tag{17}$$

Finally, the work produced by external forces is [1]:

$$W_f = \int_{\Omega} \boldsymbol{u}^T \boldsymbol{b} \, \mathrm{d}\Omega \int_{\Gamma_t} \boldsymbol{u}^T \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma \tag{18}$$

where $\boldsymbol{u}$ is the displacement. After various transformations that can be seen in detail in [1], the Galerkin weak formulation can be expressed for static problems as:

$$-\int_{\Omega} \delta\varepsilon^T \sigma \, \mathrm{d}\Omega + \int_{\Omega} \delta\boldsymbol{u}^T \boldsymbol{b} \, \mathrm{d}\Omega + \int_{\Gamma_t} \delta\boldsymbol{u}^T \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma = 0 \tag{19}$$

2.2.1. Discrete System of Equations

In FEM and meshless methods, the discrete system of equations is obtained by the principle of virtual work, using the respective shape functions as trial and test functions [1]. To begin this process, first the domain Ω is discretized into nodes. In FEM, the nodal connectivity is imposed by the nodes belonging to the same element and the vicinity of elements. In meshless methods, the nodal connectivity is established by the concept of influence-domains, or influence-cells, and the overlapping of such influence-domains [1]. The field approximation for FEM and meshless methods is given by [1]:

$$u(\boldsymbol{x}_I) = \sum_{i=1}^{n} \varphi_i(\boldsymbol{x}_I)\boldsymbol{u}_i \tag{20}$$

where $\varphi_i(\boldsymbol{x}_I)$ is the shape function of the respective numerical method and $\boldsymbol{u}_i$ are the nodal displacements of the n nodes belonging to the element or influence-domain/cell, depending on the method used, of the interest point $\boldsymbol{x}_I$. Considering the meshless methods used in this work, RPIM and NNRPIM, and the FEM, their shape functions satisfy the following condition [1], [5]:

$$\varphi_i(\boldsymbol{x}_j) = \delta_{ij} \tag{21}$$

being δ_{ij} the Kronecker delta, $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. It is important to note that even though the RPIM and NNRPIM shape functions have this property, not all meshless methods' shape functions have this property. From Equation 20, the test functions, or virtual displacements, are defined as:

$$du(\boldsymbol{x}_I) = \sum_{i=1}^{n} \varphi(\boldsymbol{x}_I) d\boldsymbol{u}_i \tag{22}$$

where $d\boldsymbol{u}_i$ are the nodal values for the test function.

2.2.2. Stiffness Matrix

The FEM and meshless formulations can be established in terms of a weak form of the differential equation under consideration, equation 19. In solid mechanics this means that the virtual work equation has to be used [1]:

$$L = \int_{\Omega} \sigma \cdot d\varepsilon \, \mathrm{d}\Omega - \int_{\Omega} d\boldsymbol{u}^T \cdot \boldsymbol{b} \, \mathrm{d}\Omega - \int_{\Gamma_t} d\boldsymbol{u}^T \cdot \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma = 0 \tag{23}$$

where the virtual work deformation, $d\varepsilon$, is defined as $d\varepsilon = \boldsymbol{B} d\boldsymbol{u}$, being $\boldsymbol{B}$ the deformation matrix. Therefore, the virtual work of the first term of Equation 23 can be expressed as

$$L_1 = \int_{\Omega} d\boldsymbol{u}^T \boldsymbol{B}^T \sigma \, \mathrm{d}\Omega \tag{24}$$

Substituting L_1 into equation 23 it is possible to write:

$$L = \int_{\Omega} d\boldsymbol{u}^T \boldsymbol{B}^T \sigma \, \mathrm{d}\Omega - \int_{\Omega} d\boldsymbol{u}^T \cdot \boldsymbol{b} \, \mathrm{d}\Omega - \int_{\Gamma_t} d\boldsymbol{u}^T \cdot \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma = 0 \tag{25}$$

For two dimensional problems with linear deformation, $\boldsymbol{B}$ can be defined as [1]:

$$\boldsymbol{B} = \begin{bmatrix} \frac{\partial \varphi_1}{\partial x} & 0 & \cdots & \frac{\partial \varphi_n}{\partial x} & 0 \\ 0 & \frac{\partial \varphi_1}{\partial y} & \cdots & 0 & \frac{\partial \varphi_n}{\partial y} \\ \frac{\partial \varphi_1}{\partial y} & \frac{\partial \varphi_1}{\partial x} & \cdots & \frac{\partial \varphi_n}{\partial y} & \frac{\partial \varphi_n}{\partial x} \end{bmatrix} \tag{26}$$

The stiffness matrix $\boldsymbol{K}$ can be determined considering the variation of the virtual work in Equation 25 in order to the generalized displacements $d\boldsymbol{u}$:

$$dL_1 = d\left[\int_{\Omega} \boldsymbol{B}^T \sigma \, \mathrm{d}\Omega\right] \tag{27}$$

and this can be developed as:

$$dL_1 = \int_{\Omega} d\boldsymbol{B}^T \sigma \, \mathrm{d}\Omega + \int_{\Omega} \boldsymbol{B}^T d\sigma \, \mathrm{d}\Omega \tag{28}$$

Knowing the constitutive relations between stress and strain and considering only linear and small deformations ($d\boldsymbol{B} = 0$), this leads to the stiffness matrix:

$$dL_1 = \int_{\Omega} \boldsymbol{B}^T \cdot (\boldsymbol{c} \cdot d\varepsilon) \, \mathrm{d}\Omega = \int_{\Omega} \boldsymbol{B}^T \cdot (\boldsymbol{c} \cdot \boldsymbol{B} d\boldsymbol{u}) \, \mathrm{d}\Omega = \int_{\Omega} \boldsymbol{B}^T \cdot \boldsymbol{c} \cdot \boldsymbol{B} d\boldsymbol{u} \, \mathrm{d}\Omega = \boldsymbol{K} \tag{29}$$

2.2.3. Force Vector

The other two integrals of Equation 23 correspond to the body forces and the external forces, respectively and their virtual work can be expressed and developed as [1]:

$$dL_2 = d\left[\int_{\Omega} d\boldsymbol{u}^T \cdot \boldsymbol{b} \, \mathrm{d}\Omega\right] = \boldsymbol{f}_b \tag{30}$$

$$dL_3 = d\left[\int_\Gamma d\boldsymbol{u}^T \cdot \bar{\boldsymbol{t}}\, \mathrm{d}\Gamma\right] = \boldsymbol{f}_{\bar{t}} \tag{31}$$

If the total force vector, $\boldsymbol{f}$, is considered to be the sum of these two forces it can be written as follows after some development:

$$\boldsymbol{f} = \int_\Omega \boldsymbol{H}^T \boldsymbol{b}\, \mathrm{d}\Omega + \int_\Gamma \boldsymbol{H}^T \bar{\boldsymbol{t}}\, \mathrm{d}\Gamma \tag{32}$$

Where $\boldsymbol{H}$ in two dimensions problems is defined as:

$$\boldsymbol{H} = \begin{bmatrix} \varphi_1 & 0 & \cdots & \varphi_n & 0 \\ 0 & \varphi_1 & \cdots & 0 & \varphi_n \end{bmatrix} \tag{33}$$

2.3. Fracture Mechanics

2.3.1. Fundamentals

Fracture Mechanics is an engineering field, related to mechanical engineering, that studies the behaviour of bodies with a crack, or multiple cracks. Fracture mechanics can be divided into many different subfields depending on the deformation and failure behaviour of the material, the type of loading and the crack behaviour [6]. The work presented here is only going to explore linear elastic fracture mechanics (LEFM) with brittle materials.

Additionally, fractures can be divided into three loading modes, mode I, mode II and mode III, which depend on how the forces are being applied to the body with the fracture. This is better exemplified in Figure 2 and the modes can be defined as follows [6]:

- Mode I: Opening Mode: The crack opens perpendicular to the crack plane. This can be caused by tensile loading.
- Mode II: In-plane sliding mode: The crack faces are displaced on their plane, normal to the crack front, which correlates to a transversal shearing load.
- Mode III: Out-of-plane tearing mode: The crack faces are displaced on their plane, parallel to the crack front, which is related to anti-plane longitudinal shearing load.

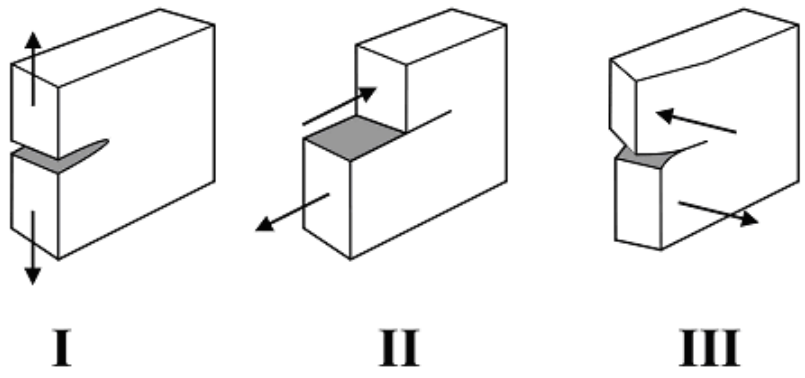

Figure 2 - Fracture Modes [7]

However, these modes represent idealised conditions and in most real-world problems the bodies are under mixed mode loading, that is, a combination of two of the modes or even all of them. This work will only approach mode I and mode II loading, isolated or mixed since this will be a 2D analysis.

The fracture mechanics field dates back to the beginning of the 20th century, when Inglis found the stresses for an elliptical crack in various configurations, in 1913 [8]. For an infinite plate with a central elliptical crack he found that:

$$\sigma_{max} = \sigma\left(1 + 2\frac{a}{b}\right) \tag{34}$$

where a and b define the shape of the ellipse as shown in Figure 3 and σ is the stress applied to the plate in infinity. This equation showed that when $b \to 0$, which is a flat crack, $\sigma_{max} \to \infty$.

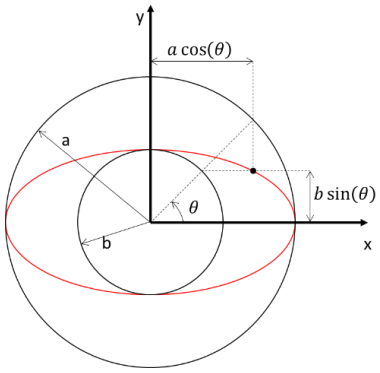

Figure 3 - Parametric representation of an ellipse

In 1920 Griffith first presented the concept of energy release [9], which is related to the Griffith's failure criterion, this is a propagation criterion that determines when the crack grows, other criterions will be discussed in Section 2.3.2.

In an infinite plate subjected to a stress, as in Figure 5, Griffith found that the strain energy would have the following relation with the crack length:

$$U = \frac{\sigma^2}{2E}V - \frac{\sigma^2}{2E}B\pi a^2 \tag{35}$$

being V the volume of the plate, B its thickness and E the Young's modulus of the material. Also considering that the energy required to break the atomic bonds and grow the crack is:

$$E_{bond} = 2\gamma_s aB \tag{36}$$

where γ_s is the energy required to break the atomic bonds per unit surface area created by the crack. Combining these two expressions, the total energy in the system is given by:

$$E_{total} = 2\gamma_s aB + \frac{\sigma^2}{2E}V - \frac{\sigma^2}{2E}B\pi a^2 \tag{37}$$

As it can be seen in Figure 4, for short crack lengths the total energy of the system increases with increasing length, meaning that to increase the crack additional energy has to be added to the system. When the crack starts to grow larger, the total energy of the system starts to decrease, leading to crack propagation without additional energy required.

This means that Equation 37 needs to be derived with respect to the crack length, a, and set it to be equal to zero to find the point where that transition occurs:

$$\frac{dE_{total}}{da} = 2\gamma_s B - \frac{\sigma^2}{2E}B\pi a = 0 \tag{38}$$

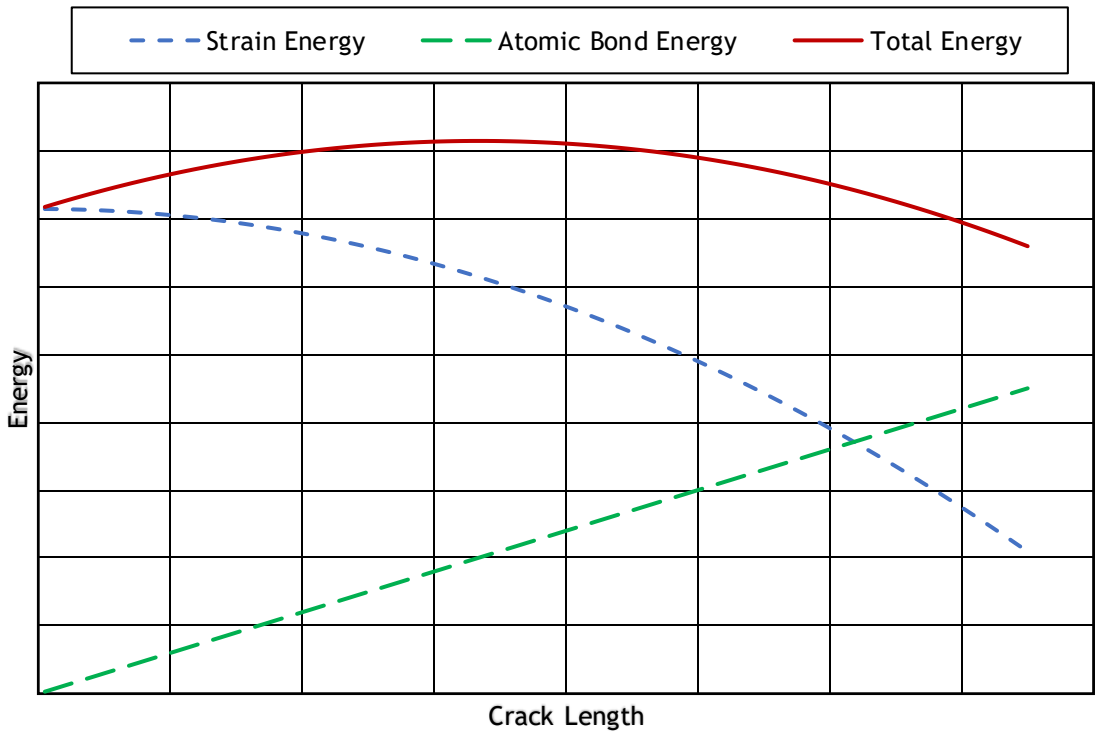

Figure 4 - Schematic representation of the energy evolution in a cracked system

It is then possible to find the amount of stress required that will lead to crack growth by solving it for σ:

$$\sigma_f = \sqrt{\frac{2\gamma_s E}{\pi a}} \tag{39}$$

where σ_f represents the failure stress. Being $2\gamma_s = G_c$ the Critical Energy Release Rate, or the Griffith's criterion, this expression can be written as:

$$\sigma_f = \sqrt{\frac{G_c E}{\pi a}} \tag{40}$$

Another very important early work in fracture mechanics are the Westergaard functions to calculate the stress fields on an infinite cracked plate [10] shown in Figure 5, they state that:

$$\sigma_{xx} = \Re eZ - y \cdot \Im mZ' \tag{41}$$
$$\sigma_{yy} = \Re eZ + y \cdot \Im mZ' \tag{42}$$
$$\sigma_{xy} = -y \cdot \Re eZ' \tag{43}$$

where

$$Z = \frac{\sigma}{\sqrt{1 - \left(\frac{a}{z}\right)^2}} \tag{44}$$

and Z' is its derivative in order to z, $z = x + iy$. When $y = 0$, $z = x$, therefore:

$$\sigma_{yy} = \Re e\left[\frac{\sigma}{\sqrt{1-\left(\frac{a}{x}\right)^2}}\right] \tag{45}$$

By looking at this equation it is possible to observe that when $x \to a$ then $\sigma_{yy} \to \infty$, which is concordant with what was found in Equation 34 when the crack is flat.

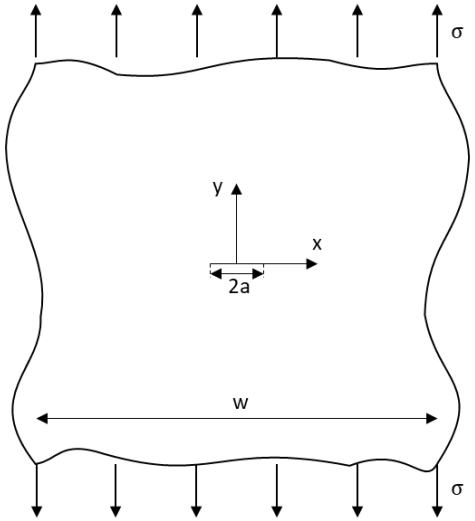

Figure 5 - Representation of the infinite plate problem

The last work that laid the foundations of fracture mechanics was Irwin's stress intensity factor (SIF) and his functions to calculate the stress fields near the crack tip [11]. Starting with the Westergaard equations of the stress fields of an infinite plate with a crack, Irwin found that by changing the coordinate system to radial coordinates with origin at the crack tip, as in Figure 6, and by assuming that $a \gg r$ for the region near the crack, it was possible to obtain the following stress fields, these are valid for the region near the crack:

$$\sigma_{xx} = \frac{K}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{46}$$

$$\sigma_{yy} = \frac{K}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1 + \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{47}$$

$$\sigma_{xy} = \frac{K}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\sin\frac{\theta}{2}\cos\frac{3\theta}{2} \tag{48}$$

where K is the stress intensity factor given by:

$$K_I = \sigma\sqrt{\pi a} \tag{49}$$

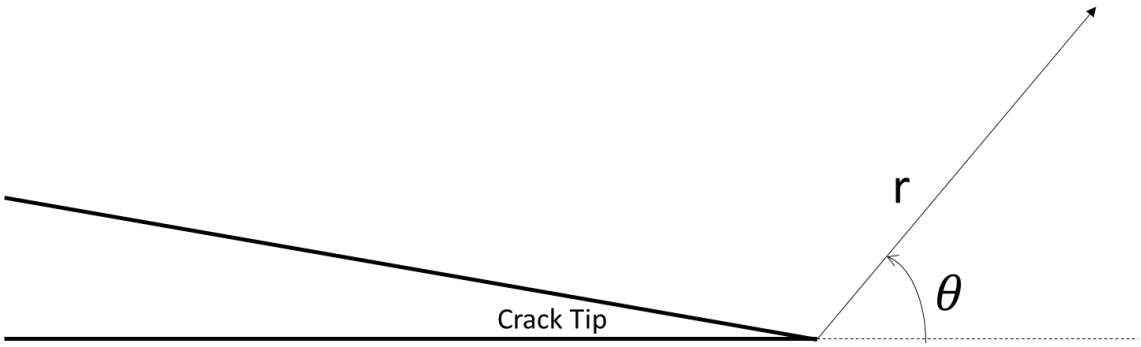

Figure 6 - Radial coordinate system

The emergence of modern computers propelled the fracture mechanics field, like most other engineering fields, because this allowed to solve more complex problems by using numerical methods. This field is usually called computational fracture mechanics, and it is explained with detail in Chapter 4.

2.3.2. Crack Propagation

The prediction of how a crack propagates is very important in fracture mechanics. There are two main features that need to be determined in this process. First, if the crack is propagating, and second, if it is, in which direction occurs the propagation.

The crack propagation criteria, can be divided in three main groups: energy based criteria, stress based criteria and strain based criteria [12].

The energy based criteria use the energy around the crack tip to determine if the crack is propagating and the direction of the propagation. This type of criteria possess more realistic assumptions regarding crack propagation than stress based criteria, because they take into consideration the energy dissipation during fracture [13]. The first energy based criterion was proposed by Griffith in 1920 [9], as mentioned before, but this is a criterion that can only be used in pure mode I conditions. According to Griffith's criterion, a fracture occurs when the stored energy in the material reaches the surface energy, however, this criterion cannot determine the crack propagation angle.

This criterion was later extended by Hussain et al. which considered the assumption that the crack propagates in the direction where the Strain Energy Release Rate (SERR) is maximum and used the criterion in a mixed mode I and II fracture [14]. Another popular energy based criteria is the Strain Energy Density (SED) criterion created by Sih [15], which considers that the crack propagates in the direction with the minimum strain energy density. This criterion was later used in many subsequent papers for both sharp V crack tips and rounded U crack tips [12].

An advantage of the stress based criteria is that they are easier to implement and understand than other criteria, the SIF introduced by Irwin can be considered one of these criteria. It can be used to determine if a crack is growing, if the SIF is higher than the fracture toughness of the material, but it does not give the propagation direction of the crack, unless it is a pure mode I problem with the crack perpendicular to the loading direction where the fracture just propagates forward.

One of the most commonly used stress based criteria is the Maximum Tangential Stress (MTS), proposed by Erdogan and Sih [16], also known as Maximum Circumferential Stress (MCS) Criterion. This is one of the simplest criteria to introduce and it gives accurate results for fracture in brittle materials. This criterion considers that the crack propagates in the

perpendicular direction to the maximum tangential stress at the crack tip. This criterion can be implement using various alternatives [17]. A commonly used implementation considers the fact that the direction of the maximum tangential stress corresponds to the first principal direction of the crack tip stress [17]. Since this criterion indicates that the crack propagates in the direction perpendicular to the maximum tangential stress, this means that crack will propagate along the second principal direction.

For some materials the strain based criteria offer a more realistic description of the fracture mechanism [13].

One example of a strain based criterion is the maximum tangential strain (MTSN) criterion [18] proposed by Chang. This criterion considers that the crack propagates in the direction of the tangential strain and reaches its maximum value, providing accurate results in both sharp crack tips and rounded crack tips.

Recently, in 2015, Mirsayar proposed an improved MTSN criterion, called EMTSN criterion [13]. This criterion considers the effects of the T-strain term in the prediction of fracture propagation.

2.3.3. J-Integral

The J-integral was originally developed by Rice [19] and Cherapanov [20] independently. Being the strain energy density W defined as:

$$W = \int_0^{\varepsilon} \sigma_{ij} d\varepsilon_{ij} \tag{50}$$

the J-integral in two dimensions was defined by Rice as:

$$J = \int_{\Gamma} \left(W dx_2 - \boldsymbol{t}_i \frac{\partial \boldsymbol{u}_i}{\partial x_1} \mathrm{ds} \right) \tag{51}$$

where x_1 and x_2 are the coordinate directions, $\boldsymbol{t}_i$ is the traction vector defined according to the outward normal along Γ, $\boldsymbol{t}_i = \sigma_{ij} \boldsymbol{n}_j$, $\boldsymbol{u}$ is the displacement vector and ds is an element of arc length along Γ. Being Γ any path encircling the crack tip, starting at the bottom face of the crack and ending at its top face. In two dimensions, for a linear elastic, homogeneous ad isotropic material, there is a relationship between the J-integral and the stress intensity factors, given by:

$$J = \frac{1}{H} (K_I^2 + K_{II}^2) \tag{52}$$

being $1/H = (1 - \nu^2)/E$ for plane strain and $1/H = 1/E$ for plane stress.

The result of the J-integral is independent of the path chosen as proved originally by Rice [19]. However in numerical applications this sometimes is not the case as shown in the results of [21]. The J-integral, can be converted into an area J-integral in 2D by using Green's theorem [22]:

$$J = \int_{\mathrm{A}} \left(\sigma_{ij} \frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_1} - W \delta_{1j} \right) \frac{\partial \mathrm{q}_1}{\partial x_{\mathrm{j}}} \mathrm{d}A \tag{53}$$

where A is the area of the annular region, q_1 is continuously differentiable within A, is unity on C_1 and vanishes on C_2

Chapter 3

Numerical Methods

3.1. Finite Element Method

The Finite Element Method (FEM) is currently the most commonly used numerical method in solid mechanics, including fracture problems. The mathematical concepts behind the FEM formulation are based on the works of Ritz, Galerkin, Trefftz and others in the beginning of the twentieth century [6]. Due to the lack of computers their work did not have many applications until the dawn of modern computer science.

The early development of FEM was conducted at University of California and it was closely linked to the aviation industry, specially Boeing, the first paper about the subject was published in 1956 by Turner et al.[23], where the FEM was applied to the structural analysis of airplane wings. This work was first presented in 1954 in meeting of the Institute of Aeronautical Sciences in New York [24]. However, in this work the method wasn't called Finite Element Method and it was Clough that later coined that term [24], which is used today.

The first application of FEM to a fracture problem can be credited to Clough in 1962 [25], in this case to the closure of a fracture due to hydrostatic loading and since then this method has been used in countless papers about fracture mechanics and many other continuum mechanics problems.

3.1.1. Mesh Generation

In 2D, to build a mesh in FEM, first the domain is divided along its x and y directions to create the nodes. With the nodes created, it is time to establish the elements, in 2D elements are triangular or quadrilateral, and they can possess distinct numbers of nodes, some examples are presented in Figure 7.

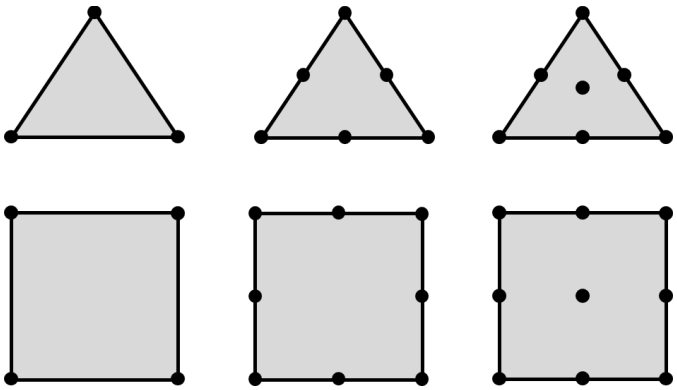

Figure 7 - Examples of different types of 2D elements

3.1.2. Integration Points

With the elements set, the next step is to create integration points for each element, this is performed using the local coordinates of each element. The number of integration points of an element is chosen accordingly to the number of nodes of the element. Figure 8 shows how the integration points of a quadrilateral element with 2x2 integration points resemble, being their local coordinates presented in Table 1.

Table 1 - Gaussian quadrature coordinates and weights

Number of points (n)	Points (x_I)	Weights (w_I)
1	0	2
2	$\pm\sqrt{\frac{1}{3}}$	1
3	0	$\frac{8}{9}$
	$\pm\sqrt{\frac{3}{5}}$	$\frac{5}{9}$
4	$\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18+\sqrt{30}}{36}$
	$\sqrt{\frac{3}{7}+\frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18-\sqrt{30}}{36}$
5	0	$\frac{128}{225}$
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$	$\frac{322+13\sqrt{70}}{900}$
	$\pm\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$	$\frac{322-13\sqrt{70}}{900}$

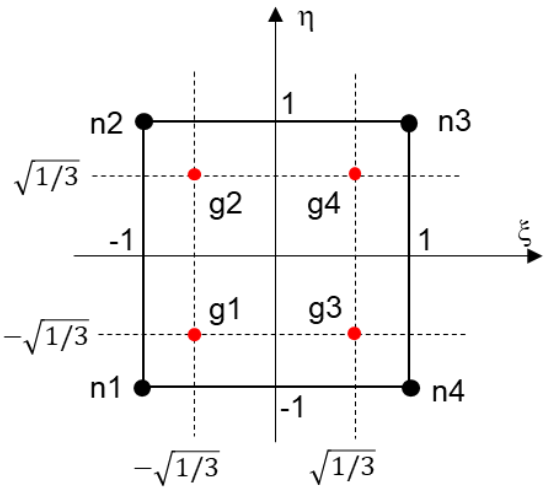

Figure 8 - Example quadrilateral "element" with 2x2 integration points

3.1.3. Shape Functions

The choice of shape function is also dependent on the user and the phenomenon to be analysed. For a more detailed explanation of the shape functions construction procedure [26] can be consulted. Considering the local coordinates of an element, ξ and η, the simplest way to define the shape functions in FEM starts by defining a matrix $\boldsymbol{C}$ of size $n \times n$, being n the number of nodes of the element. For an element with 4 nodes the $\boldsymbol{C}$ matrix can be presented as:

$$\boldsymbol{C} = \begin{bmatrix} 1 & \xi_1 & \eta_1 & \xi_1\eta_1 \\ 1 & \xi_2 & \eta_2 & \xi_2\eta_2 \\ 1 & \xi_3 & \eta_3 & \xi_3\eta_3 \\ 1 & \xi_4 & \eta_4 & \xi_4\eta_4 \end{bmatrix} \tag{54}$$

The terms of matrix $\boldsymbol{C}$ have to be chosen in accordance to the Pascal triangle, shown in Figure 9, depending on the element's number of nodes and their distribution.

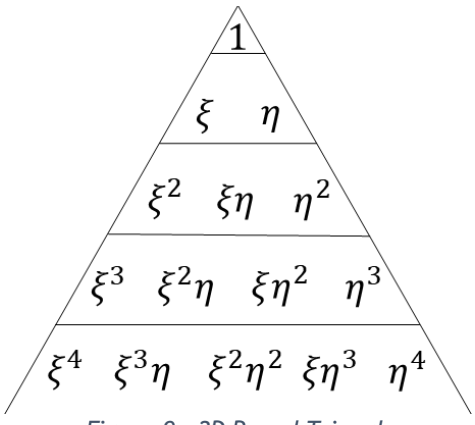

Figure 9 - 2D Pascal Triangle

If the elements are all equal this matrix will remain the same for all elements. Therefore, the shape functions of an element are defined as follows:

$$N_n = \boldsymbol{p}\boldsymbol{C}^{-1} \tag{55}$$

where $\boldsymbol{p}$ is a vector of polynomials also dependent on the Pascal triangle, with size $1 \times n$, in the example of an element with 4 nodes it would be:

$$\boldsymbol{p} = \{1 \quad \xi \quad \eta \quad \xi\eta\} \tag{56}$$

If the coordinates of a node are input in $\boldsymbol{p}$, the values of the shape function should be equal to 0 for all the shape functions except the shape function correspondent to that node, this is the Kronecker delta property.

To obtain the derivatives of the shape function in ξ and η, the polynomial has to be derived with respect to the spatial dimensions and then multiplied by the inverse of $\boldsymbol{C}$. For the given example the derivatives of the polynomial would be:

$$\begin{matrix} \partial\boldsymbol{p}/\partial\xi = \{\partial 1/\partial\xi & \xi/\partial\xi & \eta/\partial\xi & \xi\eta/\partial\xi\} = \{0 & 1 & 0 & \eta\} \\ \partial\boldsymbol{p}/\partial\eta = \{\partial 1/\partial\eta & \xi/\partial\eta & \eta/\partial\eta & \xi\eta/\partial\eta\} = \{0 & 0 & 1 & \xi\} \end{matrix} \tag{57}$$

The resulting derivatives of the shape functions would then be:

$$\begin{aligned} \partial N_n/\partial\xi &= \frac{\partial\boldsymbol{p}}{\partial\xi}\boldsymbol{C}^{-1} \\ \partial N_n/\partial\eta &= \frac{\partial\boldsymbol{p}}{\partial\eta}\boldsymbol{C}^{-1} \end{aligned} \tag{58}$$

Knowing these derivatives, it is possible to calculate the Jacobian matrix of each Gauss point by performing the following operation [1]:

$$\boldsymbol{J}_I = \begin{bmatrix} \frac{\partial N_1}{\partial\xi} & \cdots & \frac{\partial N_n}{\partial\xi} \\ \frac{\partial N_1}{\partial\eta} & \cdots & \frac{\partial N_n}{\partial\eta} \end{bmatrix} \begin{bmatrix} x_1 & x_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} \tag{59}$$

After this is done for every Gauss point $\boldsymbol{x}_I$ in the element it is possible to obtain its integration weight [1]:

$$W_I = |\boldsymbol{J}_I| w_{\xi_I} w_{\eta_I} \tag{60}$$

where w_ξ and w_η are the weights of the respective coordinate, taken from Table 1, dependent on the chosen number of integration points per axis.

The Jacobian matrix is also used to calculate the derivatives of the shape function in order to x and y at each Gauss point:

$$\begin{bmatrix} \frac{\partial N_1}{\partial x} & \cdots & \frac{\partial N_n}{\partial x} \\ \frac{\partial N_1}{\partial y} & \cdots & \frac{\partial N_n}{\partial y} \end{bmatrix} = \boldsymbol{J}_I^{-1} \begin{bmatrix} \frac{\partial N_1}{\partial\xi} & \cdots & \frac{\partial N_n}{\partial\xi} \\ \frac{\partial N_1}{\partial\eta} & \cdots & \frac{\partial N_n}{\partial\eta} \end{bmatrix} \tag{61}$$

The last use of the shape functions in FEM is to convert the coordinates of the Gauss points from their local coordinates to the global coordinates:

$$\{x_I \quad y_I\} = [N_1 \quad N_2 \quad N_3 \quad N_4] \cdot \begin{bmatrix} x_{n1} & y_{n1} \\ x_{n2} & y_{n2} \\ x_{n3} & y_{n3} \\ x_{n4} & y_{n4} \end{bmatrix} \tag{62}$$

3.1.4. Stiffness Matrix

In FEM, each element has its own local stiffness matrix and then there is a global stiffness matrix, which results from the assembly of all local stiffness matrices. To calculate the local stiffness matrix of one element, the first step is to create the deformation matrix $\boldsymbol{B}$ of each Gauss point, which has the following structure [1]:

$$\underbrace{\boldsymbol{B}_I}_{[3\times 2n]} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \cdots & \frac{\partial N_n}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & \cdots & 0 & \frac{\partial N_n}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \cdots & \frac{\partial N_n}{\partial y} & \frac{\partial N_n}{\partial x} \end{bmatrix} \tag{63}$$

By knowing the deformation matrix for a Gauss point, it is possible to calculate the stiffness matrix for that Gauss point, to do this the constitutive matrix, which is the inverse of equation 10, 12 or 13 depending on the theory used, is also needed.

Then the local stiffness matrix for the integration point I is given by [1]:

$$\underbrace{\boldsymbol{K}_I}_{[2n\times 2n]} = W_I \cdot [\boldsymbol{B}_I]^T \cdot \boldsymbol{c} \cdot \boldsymbol{B}_I \tag{64}$$

This will give the stiffness for the nodes of the element corresponding to that integration point. Finally, the global stiffness matrix is constructed by assembling (U) the stiffness matrices of all the integration points:

$$\underbrace{\boldsymbol{K}_{global}}_{[2N\times 2N]} = \bigcup_{I=1}^{Q} \boldsymbol{K}_I \tag{65}$$

where Q is the total number of integration points of the domain and N is the total number of nodes in domain.

3.1.5. Natural Boundary Conditions

To apply a force along a boundary line Γ the nodes on that boundary have to be identified, $\boldsymbol{X}_\Gamma$. Then a new set of integration points along the boundary must be created, Q_Γ, because the integration points discretizing the domain cannot be used to integrate a function along the boundary line.

The process to obtain the shape function values at the integration points is the same as described in 3.1.3, but in this case the elements of the boundary line are 1D, not 2D.

The shape functions are then used to create the matrix $\boldsymbol{H}$ for the integration point I:

$$\underbrace{\boldsymbol{H}_I}_{[2\times 2n]} = \begin{bmatrix} \varphi_1 & 0 & \cdots & \varphi_n & 0 \\ 0 & \varphi_1 & \cdots & 0 & \varphi_n \end{bmatrix} \tag{66}$$

The boundary external force vector can then be calculated using the following summation [1]:

$$\boldsymbol{f}_e = \sum_{I=1}^{Q} w_I^\Gamma \, \boldsymbol{H}_I^T \, \bar{\boldsymbol{t}_c}(\boldsymbol{q}_I) \tag{67}$$

where $\bar{\boldsymbol{t}}_c(\boldsymbol{q}_I)$ is a vector dependent on the position of the integration point, composed by the function that defines the forces applied to the boundary in the all the Cartesian directions.

3.1.6. Essential Boundary Conditions

There are different ways to impose the essential boundary conditions, the most common methods are the "penalty method" or the "direct imposition method". In this work, the direct imposition method was used, so it is the method described in this section, for a detailed description of the penalty method [1] can be consulted.

Using the direct imposition method, the essential boundary conditions can be implemented directly into the system of equations $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$ by modifying both the global stiffness matrix and the global force vector.

Being X_Γ a set of nodes belonging to a boundary where the user wants to impose essential boundary conditions. Consider a general case where each field node $\boldsymbol{x}$ has m degrees of freedom. The field node $\boldsymbol{x}_I \in \boldsymbol{X}_\Gamma$ has a displacement constrain $\bar{\boldsymbol{u}}$ on the J^{th} degree of freedom. As in the natural boundary conditions, the first step here is to identify the nodes in the boundary. Then the essential boundary condition can be imposed by performing these modifications to the stiffness matrix and the force vector [1]:

$$K_{(m\cdot i-(m-n)),(m\cdot j-(m-k))} = \begin{cases} 0 & if\ i = I\ \wedge n = J \\ 1 & if\ i = j = I\ \wedge n = k = J \\ K_{(m\cdot i-(m-n))(m\cdot j-(m-k))} & if\ i \neq I \vee (i = I\ \wedge n \neq J) \end{cases} \tag{68}$$

$$f_{(m\cdot i-(m-n))} = \begin{cases} \bar{u} & if\ i = I\ \wedge n = J \\ f_{(m\cdot i-(m-n))} & if\ i \neq I \vee (i = I\ \wedge n \neq J) \end{cases} \tag{69}$$

where $\{i, j\} = \{1,2,\dots,N\}$ and $\{n, k\} = \{1,2,\dots,m\}$.

3.1.7. Strain, Stress and Displacement

The displacement vector, $\boldsymbol{u}$, is calculated by the following formula:

$$\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{f} \tag{70}$$

being $\boldsymbol{K}$ the stiffness matrix with the imposed constrains and $\boldsymbol{f}$ the force vector with the imposed constrains and the applied external forces.

Then the strain is calculated for each integration point using the following formula:

$$\varepsilon_I = \boldsymbol{B}_I\boldsymbol{u}_e \tag{71}$$

where $\boldsymbol{B}_I$ is the matrix from equation 63 and $\boldsymbol{u}_e$ is the displacement vector of the element to which that integration point belongs. From equation 14, the relationship between the stress and the strains at a Gauss point is:

$$\sigma_I = \boldsymbol{D}\varepsilon_I \tag{72}$$

In the 2D case, depending on if either plane stress or plane strain is used, σ_{zz} or ε_{zz} have to be calculated using equation 14.

3.2. Meshless Methods

Meshless methods are discrete numerical methods used to solve partial differential equations (PDEs), these methods offer some advantages when compared to the more common FEM, one of them is due to the fact that they do not possess a mesh. They can solve simulations of failure processes, such as the propagation of cracks with more ease, because there is no need for remeshing [3], [27]. In contrast with FEM, when using meshless methods the influence domains can and should overlap each other. Many different types of meshless methods exist, they can differ in the interpolation or approximation function, the used formulation and the integration [1].

There are many interpolation or approximation functions that can be used. Some of the most common are the Taylor approximation, the moving least-square approximation, the hp-cloud approximation function, the polynomial interpolation and the radial interpolation [1]. The interpolation, or approximation, function needs a domain of applicability, outside this domain the values of the function will be zero, this domain is called influence-domain and it is comparable to the element in FEM [1].

Regarding the formulation, there are two different types, strong formulation and weak formulation. The strong formulation uses the partial differential equations that govern the problem directly without approximations to obtain the solution; while the weak formulation uses a variational principle to minimize the residual weight of the differential equations governing the problem. The residual is obtained through the substitution of the exact solution by an approximated function affected by a test function [1]. Since many differential equations that govern real phenomena do not admit sufficiently smooth solutions, weak formulations are very important, because they allow to find the solutions for these types of problems [3], being this type of formulation very important in solid mechanics[3].

The first use of meshless methods was the smoothed particle hydrodynamics (SPH) in 1977 by Lucy [28] and Gingold and Monaghan [29]. This method was originally used to model astrophysics problems, a method that was further improved over the years [2] and later started to gain attention in the solid mechanics field to solve problems that were difficult to solve using the more common mesh based methods, such as fracture or impact problems [30]. The SPH had some flaws (such as low accuracy and instabilities), which led to the improvement of this method in subsequent papers [31], [32].

The SPH and its improvements are based on the strong formulation of the partial differential equations (PDE), but in the 1990s a different approach to solving the PDEs in meshless methods appeared by using the weak formulation of the PDEs. The commonly used moving least squares (MLS) approximants were introduced in 1981 by Lancaster and Salkauskas [33] and the first meshless method to use these approximants to build the approximation functions was the diffuse element method (DEM) created by Nayroles, Touzot and Villon in 1992 [34], being also the first meshless method to use the weak formulation.

Introduced in 1994 by Belytschko et al. the element-free Galerkin (EFG) method [35] was an improvement on the DEM, this method was used one year later to solve crack propagation problems [36] and it was also the first meshless methods, coupled with FEM, to solve some of the problems with the meshless formulation, being this mixed method also used in fracture growth problems [37]. Around the same time, Liu et al. introduced the reproducing kernel particle method (RKPM) [38], which uses the reproducing kernel approximant, this method was shown to be an improvement over DEM and SPH.

There are also some methods that use local weak forms instead of global weak forms, one of the most popular of these methods is the meshless local Petrov-Galerkin (MLPG) method [39], where the integration of the weak form is performed in the local domains.

The meshless methods presented until here use approximants and they achieve good solutions for solid mechanics problems, but they have some issues that need to be taken into account to obtain good solutions. One of them, and possibly the most significant one, is their lack of the Kronecker delta property, which makes the imposition of essential and natural boundary conditions difficult. There have been some works proposing a solution for this issue, such as the use of Nitche's method [40] applied to meshless methods [41], or the modification of the approximation space near the boundary [42].

One of the most important solutions, in the context of this work, to solve the boundary enforcement issue was the natural element method (NEM) [43], [44], which was one of the first meshless methods to use interpolant functions. This method uses geometrical and mathematical concepts, such as the Voronoï diagrams[45] and the Delaunay tessellation[46], to construct the interpolation functions.

Another important method in the context of this work is the point interpolation method (PIM) [47] and the point assembly method [48], which were introduced by Liu. These methods also use interpolant functions, instead of approximants, so they have the Kronecker delta property, making it easier (and as straightforward as in the FEM) to impose essential boundary conditions. Liu and Wang later addressed some of the problems with the PIM when they developed the radial point interpolation method (RPIM) [5], which uses radial point interpolation functions instead of the original polynomial functions. This is one of the meshless methods used in this work.

The other meshless method used in this work is the natural neighbour radial point interpolation method (NNRPIM) [49], [50] developed at FEUP and introduced in 2007 by Dinis, Renato Natal and Belinha. It is a method inspired by both the NEM and the RPIM.

This method uses the concept of "influence cell", as opposed to the "influence domain" of the majority of meshless methods. To obtain the influence cells the NNRPIM uses the Voronoï diagrams and the Delaunay tessellation, similarly to the NEM. Using the Voronoï cells, a set of influence cells is created departing from an unstructured set of nodes. The Delaunay triangles are used to create the background mesh, used in the integration of the interpolation functions. Since this background mesh is nodal dependent, the NNRPIM can be considered a truly meshless method.

The NNRPIM was already used in many different computational mechanics fields, including fracture mechanics problems[4], [51], [52], which are of special interest to this work.

3.2.1. Node Generation

The creation of nodes in meshless methods is very similar to the mesh creation in FEM, in section 3.1.1, but in this case the elements are not created, just the nodes. So, the domain is divided along its directions, in 2D x and y, and nodes are created along those division, just like in the FEM.

3.2.2. Influence Domains

Since unlike in the FEM here there are no elements, the nodal connectivity is achieved by the influence domains of interest points and the overlapping of those influence domains. Commonly, these interest points are the integration points, and most meshless methods use a

background mesh to generate the integration points, this is a simple and effective way to generate them. This background integration mesh can be dependent or independent from the nodes. This provides some flexibility in the generation of integration points and differs from the FEM where the integration points are dependent on the mesh. The size of the integration mesh does not have a significant effect in the final results [35].

When using this background integration mesh, commonly the Gaussian quadrature is used, and the integration points creation is very similar to the one shown for FEM in Section 3.1.2.

The influence-domain of an integration point can be defined by a fixed area, having the shape of a square or a circle and the influence-domain of each integration will have the same size, but a different number of nodes inside of it; or it can be defined by the number of nodes inside it, doing a radial search around the integration point to find the nodes closer to it, making the size of the influence-domain of each integration point different, but having the same number of nodes inside of it.

Due to its simplicity, the area defined influence domain is commonly used. However, having a significant difference in the number of nodes in the influence-domains affects the performance and the final solution of the meshless method, so it is important to have approximately the same number of nodes in all influence-domains [1].

Independently of the meshless method used the number of nodes inside a influence domain should be between 9 and 16 [1], [47], [53].

3.2.3. Natural Neighbours and Influence Cells

Another approach is to use influence cells instead of influence domains. This is achieved by using Voronoï diagrams and Delaunay triangulation and constitutes the basis of the NNRPIM. This is a concept that was first proposed by Dinis, Natal Jorge and Belinha [49].

This theory can be used in a $\mathcal{D}$-dimensional space, but since the work presented here is for 2D problems, the theory behind this concept is presented for a 2D Euclidian space $\mathbb{R}^2$. Consider a set $\boldsymbol{N}$ of N distinct nodes, discretizing the domain $\Omega \subset \mathbb{R}^2$:

$$\boldsymbol{N} = \{n_1, n_2, \dots, n_N\} \in \mathbb{R}^2 \tag{73}$$

The Voronoï diagram of $\boldsymbol{N}$ is the partition of the domain defined by this set into sub-regions V_i, closed and convex, being each of those sub-regions associated with the node i, n_i, such that any point inside V_i is closer to n_i than any other node $n_j \in N \wedge j \neq i$. The set of these sub-regions $V = \{V_1, V_2, \dots, V_N\}$ defines the Voronoï diagram. Mathematically the Voronoï cell is defined by [1]:

$$V_i = \left\{\boldsymbol{x}_I \in \Omega \subset \mathbb{R}^2 : \|\boldsymbol{x}_I, \boldsymbol{x}_i\| < \left\|\boldsymbol{x}_I, \boldsymbol{x}_j\right\|, \quad \forall\, i \neq j\right\} \tag{74}$$

Where $\boldsymbol{x}_I$ is an interest point of the domain and $\|\cdot\|$ is the Euclidian metric norm. To visualise how a Voronoï cell is built, consider the set of nodes in Figure 10a. To find the Voronoï cell of node n_o, first one of the other nodes has to be selected as a potential neighbour, if node n_4 is selected, as in Figure 10b, then the normal vector u_{40} given by:

$$u_{40} = \frac{x_0 - x_4}{\|x_0 - x_4\|} \tag{75}$$

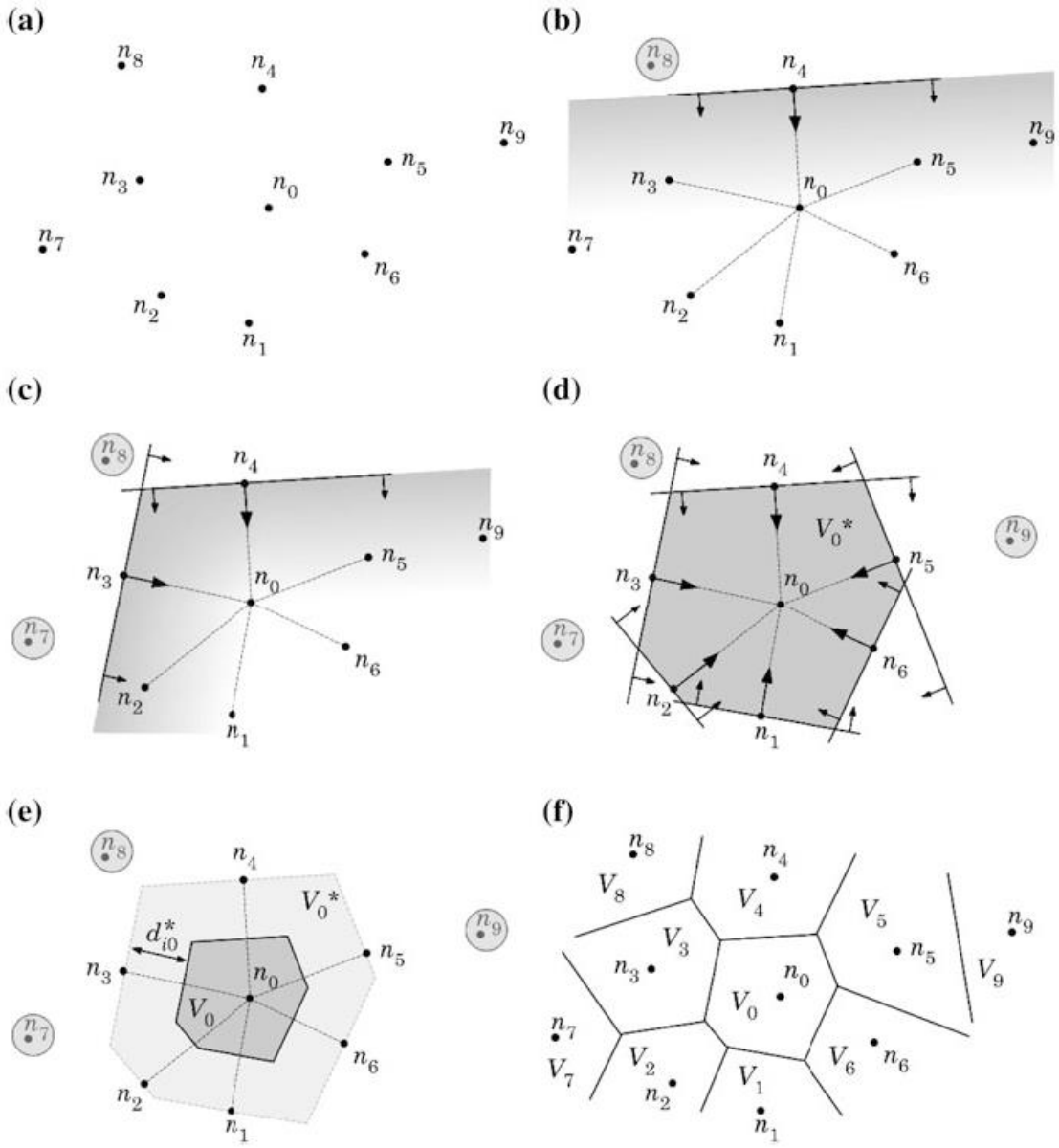

Figure 10 – (a) Initial nodal set of potential neighbour nodes of node n_0. (b) First trial plane. (c) Second trial plane. (d) Final trial cell containing just the natural neighbours of node n_0. (e) Node n_0 Voronoï cell V_0. (f) Voronoï diagram [1]

The components of $u_{40} = \{u_{40}, v_{40}, w_{40}\}$, with them it is possible to define the plane π_{40}:

$$u_{40}x + v_{40}y + w_{40}z = (u_{40}x_4 + v_{40}y_4 + w_{40}z_4) \tag{76}$$

With this plane defined, it is possible to exclude as natural neighbours all the nodes that not respect the following condition [1]:

$$u_{40}x + v_{40}y + w_{40}z \leq (u_{40}x_4 + v_{40}y_4 + w_{40}z_4) \tag{77}$$

As seen in Figure 10b this resulted in the exclusion of n_8 as a natural neighbour of n_0, this process is then repeated using another node, in Figure 10c n_3 is used, and it is repeated like that for every node until Figure 10d is reached which contains all the natural neighbours of n_0 which respect all the following 6 conditions [1]:

$$\begin{cases} u_{10}x + v_{10}y + w_{10}z \leq (u_{10}x_1 + v_{10}y_1 + w_{10}z_1) \\ u_{20}x + v_{20}y + w_{20}z \leq (u_{20}x_2 + v_{20}y_2 + w_{20}z_2) \\ \vdots \\ u_{60}x + v_{60}y + w_{60}z \leq (u_{60}x_6 + v_{60}y_6 + w_{60}z_6) \end{cases} \tag{78}$$

Only the nodes on the perimeter of the domain V_0^* are considered neighbour nodes. The Voronoï cells of n_0 can then be seen in Figure 10e and it is the homothetic form of the auxiliary domain V_0^*, where:

$$d_{0i}^* = \frac{d_{0i}}{2} = \frac{\|x_0 - x_i\|}{2} \tag{79}$$

The same process is then used to determine the Voronoï cells of the rest of the nodes in the domain and the result is seen in Figure 10f.

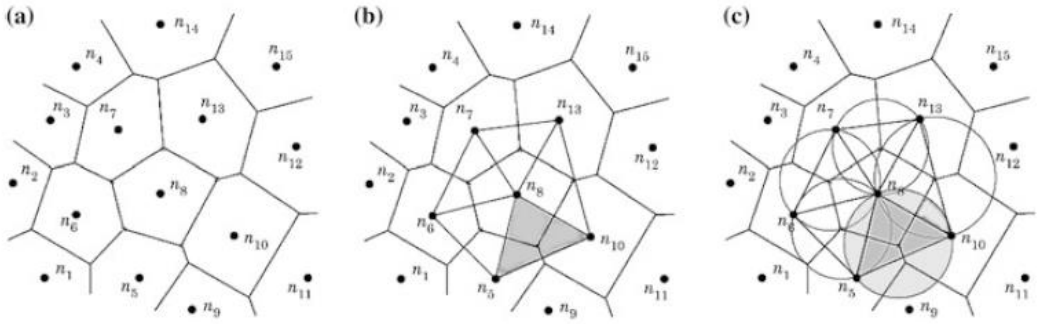

Figure 11 - (a) Voronoï diagram (b) Delaunay Triangulation (c) Natural neighbour circumcircle [1]

The Delaunay triangulation is the geometrical dual of the Voronoï diagram and it is built by connecting the nodes that have Voronoï cells with common boundaries. The duality between the Voronoï diagram and the Delaunay triangulation means that there is a Delaunay triangle between two nodes in a plane, if and only if their Voronoï cells share a common edge. An important property of the Delaunay triangles is the "empty circumcircle criterion" [1]. If a set of nodes $N_t = \{n_j, n_k, n_l\} \in \mathbf{N}$ forms a Delaunay triangle, then the circumcircle formed by that triangle contains only the nodes of N_t and no other nodes of the global set $\mathbf{N}$. The centre of this circumcircle is the vertex that is shared by all Voronoï cells of N_t. These properties of the Delaunay triangulation are shown in Figure 11.

In the NNRPIM the Voronoï diagram is used to create the "influence-cells", which enforce the nodal connectivity between the nodes that discretize the domain. In this method there are two degrees of influence-cells.

The "first degree influence-cell" of an interest point $\boldsymbol{x}_I$ is found by searching for its natural neighbours, using the Natural Neighbour Voronoï construction shown before, therefore the first degree influence-cell for that point would be the one shown in Figure 12a [1].

The "second degree influence-cell" of an interest point $\boldsymbol{x}_I$ is found by following the same procedure to find the neighbour nodes of the point and then the natural neighbours of the first natural neighbours are also added to the influence-cell, as shown in Figure 12b [1].

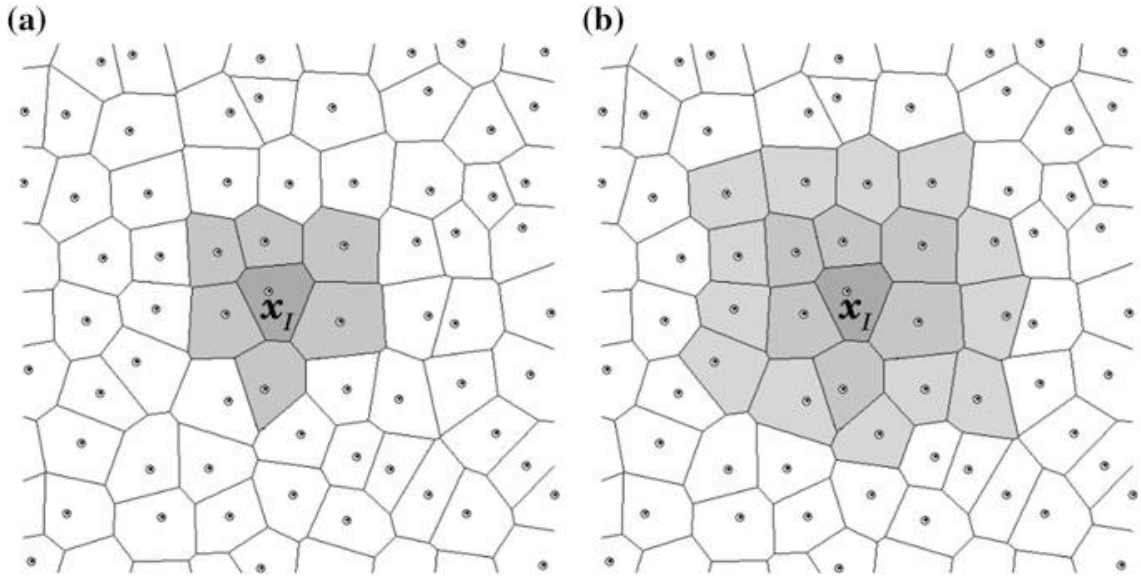

Figure 12 – (a) 1st degree influence-cells (b) 2nd degree influence-cells [1]

With the natural neighbours concept, the creation of integration points is related to the Voronoï cells of each node, this means that the integration points of a node only belong to that node and no other node.

Using the Voronoï cells and the Delaunay triangulation, it is possible to subdivide each cell into small sub cells, which are quadrilaterals if the nodes are irregularly distributed, as in Figure 13, and are triangles if the nodes are regularly distributed, as in Figure 14 [1].

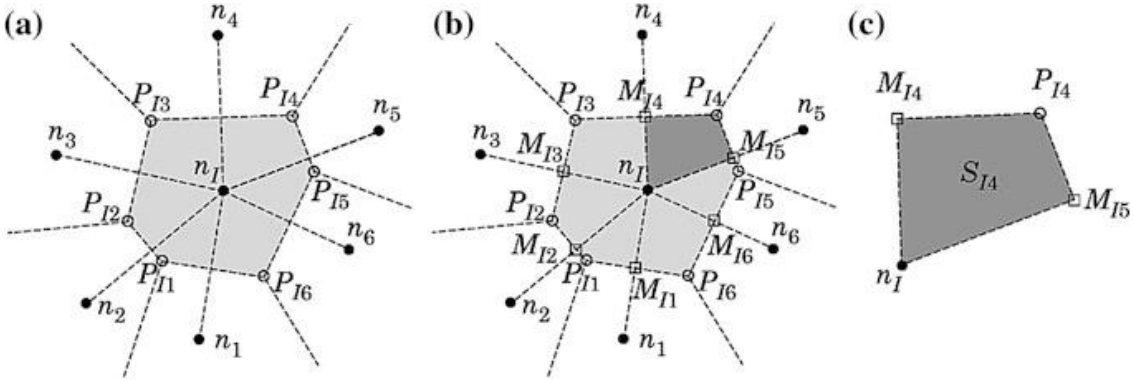

Figure 13 – Cell sub divisions with irregularly distributed nodes [1]

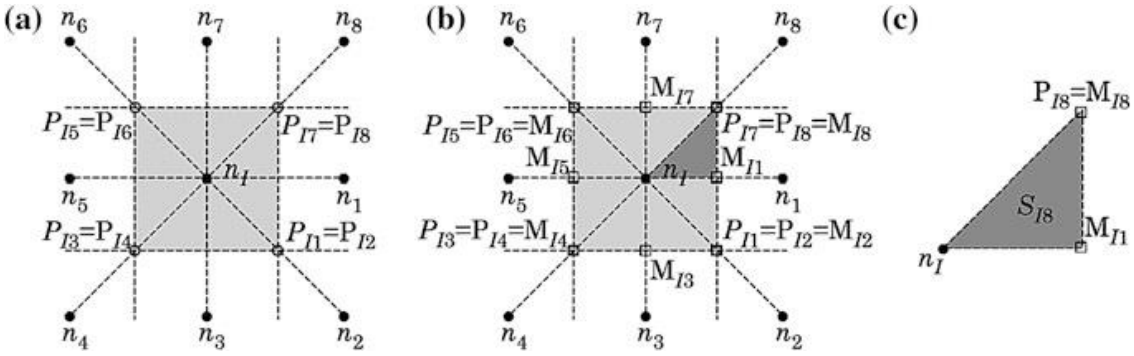

Figure 14 - Cell sub divisions with regularly distributed nodes [1]

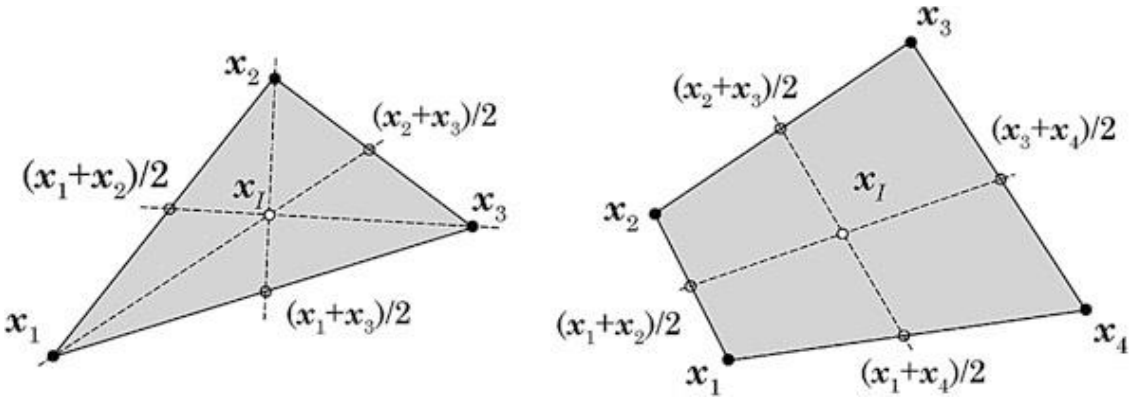

Figure 15 - Integration point location in the sub cells [1]

The simplest way to create the integration points of a sub cells is to simply create one at its barycentre, following the scheme presented in Figure 15. Using this technique to create the integration points, the weight of each integration point would be equal to the area of its respective sub cell. The area of a sub cell can be defined as follows for a triangle:

$$A_I = \frac{1}{2}\left|\det\begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix}\right| \tag{80}$$

And for a quadrilateral the area is defined as:

$$A_I = \frac{1}{2}\left|\det\begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix} + \det\begin{bmatrix} x_4 - x_1 & y_4 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix}\right| \tag{81}$$

Another, more general, way to create integration points is to subdivide each sub-cell into quadrilaterals and then use the Gaussian-Legendre quadrature to create integration points for each of those subdivisions, as seen in Figure 16.

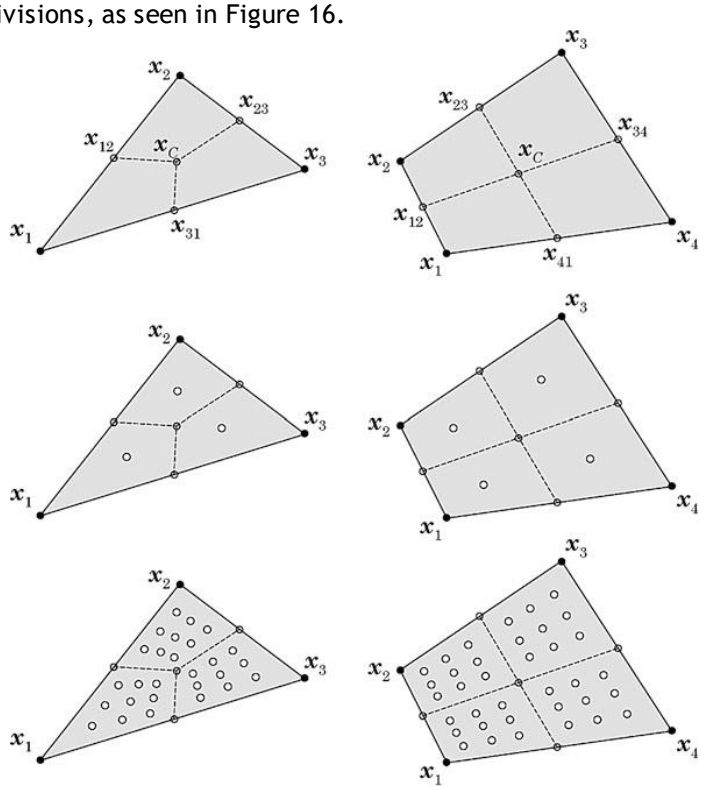

Figure 16 - Integration points of a sub-cell using the Gauss-Legendre integration scheme

3.2.1. Shape functions

In the RPIM and the NNRPIM the shape function construction is more complex than in the FEM. Considering $u(\boldsymbol{x})$, a function in the domain $\Omega \subset \mathbb{R}^{\mathrm{d}}$ discretized by a set of nodes N with coordinates $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ it is possible to define for an interest point $\boldsymbol{x}_I \in \mathbb{R}^d$, not necessarily coincident with $\boldsymbol{X}$, that defines the coordinates of a set of nodes N, the radial point interpolation function $u(\boldsymbol{x}_I)$ as:

$$u^h(\boldsymbol{x}_I) = \sum_{i=1}^{n} r_i(\boldsymbol{x}_i - \boldsymbol{x}_I) a_i(\boldsymbol{x}_I) + \sum_{j=1}^{m} p_j(\boldsymbol{x}_I) b_j(\boldsymbol{x}_I) = r(\boldsymbol{x}_I)^T a(\boldsymbol{x}_I) + p(\boldsymbol{x}_I)^T b(\boldsymbol{x}_I) \tag{82}$$

With $a_i(\boldsymbol{x}_I)$ and $b_j(\boldsymbol{x}_I)$ being the non-constant coefficient of $r(\boldsymbol{x}_I)$ and $p(\boldsymbol{x}_I)$ respectively, that can be defined as:

$$\boldsymbol{a}(\boldsymbol{x}_I) = \{a_1(\boldsymbol{x}_I) \quad a_2(\boldsymbol{x}_I) \quad \cdots \quad a_n(\boldsymbol{x}_I)\}^T \tag{83}$$
$$\boldsymbol{b}(\boldsymbol{x}_I) = \{b_1(\boldsymbol{x}_I) \quad b_2(\boldsymbol{x}_I) \quad \cdots \quad b_m(\boldsymbol{x}_I)\}^T \tag{84}$$

Being n the number of nodes inside the influence cell of the interest point $\boldsymbol{x}_I$ and m the number of monomial of the complete polynomial basis $p_j(\boldsymbol{x}_I)$, that can be defined with the help of the Pascal triangle, Figure 9, having the following vector form:

$$\boldsymbol{p}(\boldsymbol{x}_I) = \{p_1(\boldsymbol{x}_I) \quad p_2(\boldsymbol{x}_I) \quad \cdots \quad p_m(\boldsymbol{x}_I)\}^T \tag{85}$$

The radial basis function (RBF) can be defined as:

$$\begin{aligned} \boldsymbol{r}(\boldsymbol{x}_I) &= \{r_1(\boldsymbol{x}_I) \quad r_2(\boldsymbol{x}_I) \quad \cdots \quad r_n(\boldsymbol{x}_I)\}^T \\ &= \{r(\boldsymbol{x}_1 - \boldsymbol{x}_I) \quad r(\boldsymbol{x}_2 - \boldsymbol{x}_I) \quad \cdots \quad r(\boldsymbol{x}_n - \boldsymbol{x}_I)\}^T \end{aligned} \tag{86}$$

The only variable in the RBF is the Euclidean norm between the nodes and the integration point, d_{iI}, which can be defined, for a two-dimensional space, as:

$$d_{iI} = \sqrt{(x_i - x_I)^2 + (y_i - y_I)^2} \tag{87}$$

There are many different RBF that can be incorporated into the RPI formulation [1], [5], [53]. The most commonly used globally supported RBFs are the multi-quadrics (MQ) function, the Gaussian function and the thin plate spline function [1]. Here the used RBF was the multiquadric function which can be seen bellow:

$$r_i(x_I) = (d_{il}^2 + (\gamma d_a)^2)^p \tag{88}$$

being γ and p shape parameters and $d_a = w_I$, the weight of the interest point x_I.

According to Liu and Wang the shape parameters γ and p should be 1.03 and 1.42 using the RPIM [53]. But using the NNRPIM formulation Dinis et al. found that the optimal values for the shape parameters are $\gamma \leq 0.0001$ and $p \cong 1$ [49]. It is however important to note that γ must not be 0 because it leads to ill conditioned or singular moment matrices and p must not be equal to 1 or any other integer value, because it will make the moment matrix singular[1].

To obtain the non-constant coefficients $\boldsymbol{a}(\boldsymbol{x}_I)$ and $\boldsymbol{b}(\boldsymbol{x}_I)$, $u^h(\boldsymbol{x}_I)$ is imposed to pass through all the nodal values n of the support-domain of $\boldsymbol{x}_I$, obtaining the following system of equations, expressed in matrix form:

$$\boldsymbol{R}\,\boldsymbol{a}(\boldsymbol{x}_I) + \boldsymbol{P}\,\boldsymbol{b}(\boldsymbol{x}_I) = \boldsymbol{u}_s \tag{89}$$

Being $\boldsymbol{u}_s$ the vector containing the nodal parameters of the field function for each node inside the support domain of the RPI shape function defined as:

$$\boldsymbol{u}_s^T = \{u_1 \quad u_2 \quad \cdots \quad u_n\} \tag{90}$$

For the MQ RBF the radial moment matrix is defined as:

$$\boldsymbol{R} = \begin{bmatrix} (d_{11}^2 + (\gamma d_c)^2)^p & (d_{12}^2 + (\gamma d_c)^2)^p & \cdots & (d_{1n}^2 + (\gamma d_c)^2)^p \\ (d_{21}^2 + (\gamma d_c)^2)^p & (d_{22}^2 + (\gamma d_c)^2)^p & \cdots & (d_{2n}^2 + (\gamma d_c)^2)^p \\ \vdots & \vdots & \ddots & \vdots \\ (d_{n1}^2 + (\gamma d_c)^2)^p & (d_{n2}^2 + (\gamma d_c)^2)^p & \cdots & (d_{nn}^2 + (\gamma d_c)^2)^p \end{bmatrix} \tag{91}$$

And the polynomial moment matrix is defined as:

$$\boldsymbol{P} = \begin{bmatrix} p_1(\boldsymbol{x}_1) & p_2(\boldsymbol{x}_1) & \cdots & p_m(\boldsymbol{x}_1) \\ p_1(\boldsymbol{x}_2) & p_2(\boldsymbol{x}_2) & \cdots & p_m(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\boldsymbol{x}_n) & p_2(\boldsymbol{x}_n) & \cdots & p_m(\boldsymbol{x}_n) \end{bmatrix} \tag{92}$$

To obtain a unique solution another set of equations has to be used, which can be written in matrix form as:

$$\boldsymbol{P}^T\,\boldsymbol{a}(\boldsymbol{x}_I) = 0 \tag{93}$$

Combining Equation 89 and Equation 93, it is possible to obtain the following set of equations, written in matrix form:

$$\begin{bmatrix} \boldsymbol{R} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{Z} \end{bmatrix} \begin{Bmatrix} \boldsymbol{a}(\boldsymbol{x}_I) \\ \boldsymbol{b}(\boldsymbol{x}_I) \end{Bmatrix} = \boldsymbol{M}_T \begin{Bmatrix} \boldsymbol{a}(\boldsymbol{x}_I) \\ \boldsymbol{b}(\boldsymbol{x}_I) \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{u}_s \\ \boldsymbol{z} \end{Bmatrix} \tag{94}$$

Where $Z_{ij} = 0$ for $\{i, j\} = 1, 2, \dots, m$ and $z_i = 0$ for $i = 1, 2, \dots, m$. To obtain $\boldsymbol{a}(\boldsymbol{x}_I)$ and $\boldsymbol{b}(\boldsymbol{x}_I)$:

$$\begin{Bmatrix} \boldsymbol{a}(\boldsymbol{x}_I) \\ \boldsymbol{b}(\boldsymbol{x}_I) \end{Bmatrix} = \boldsymbol{M}_T^{-1} \begin{Bmatrix} \boldsymbol{u}_s \\ \boldsymbol{z} \end{Bmatrix} \tag{95}$$

This can be substituted into Equation 82, obtaining the following equation:

$$u^h(\boldsymbol{x}_I) = \{\boldsymbol{r}(\boldsymbol{x}_I)^T \quad \boldsymbol{p}(\boldsymbol{x}_I)^T\}\,\boldsymbol{M}_T^{-1} \begin{Bmatrix} \boldsymbol{u}_s \\ \boldsymbol{z} \end{Bmatrix} \tag{96}$$

Since the field variable value for an interest point $\boldsymbol{x}_I$ is interpolated using the shape function values obtained at the nodes inside the influence cell of the interest point, the interpolation function vector $\varphi(\boldsymbol{x}_I)$, with size n, can be identified on Equation 96:

$$u^h(\boldsymbol{x}_I) = \{\boldsymbol{r}(\boldsymbol{x}_I)^T \quad \boldsymbol{p}(\boldsymbol{x}_I)^T\}\,\boldsymbol{M}_T^{-1} \begin{Bmatrix} \boldsymbol{u}_s \\ \boldsymbol{z} \end{Bmatrix} = \{\varphi(\boldsymbol{x}_I)^T \quad \psi(\boldsymbol{x}_I)^T\} \begin{Bmatrix} \boldsymbol{u}_s \\ \boldsymbol{z} \end{Bmatrix} \tag{97}$$

Where $\psi(\boldsymbol{x}_I)^T$ is a by-product vector that does not possess any relevant meaning or data. Therefore, the interpolation function vector $\varphi(\boldsymbol{x}_I)^T$ can be defined as:

$$\{\varphi(\boldsymbol{x}_I)^T \quad \psi(\boldsymbol{x}_I)^T\} = \{\boldsymbol{r}(\boldsymbol{x}_I)^T \quad \boldsymbol{p}(\boldsymbol{x}_I)^T\}\, \boldsymbol{M}_T^{-1} \tag{98}$$

The first order partial derivative of this function in respect to a generic variable ξ can be written as follows:

$$\left\{\frac{\partial\varphi(\boldsymbol{x}_I)^T}{\partial\xi} \quad \frac{\partial\psi(\boldsymbol{x}_I)^T}{\partial\xi}\right\} = \left\{\frac{\partial\boldsymbol{r}(\boldsymbol{x}_I)^T}{\partial\xi} \quad \frac{\partial\boldsymbol{p}(\boldsymbol{x}_I)^T}{\partial\xi}\right\} \boldsymbol{M}_T^{-1} \tag{99}$$

Being the MQ partial derivative:

$$\frac{\partial r_i(\boldsymbol{x}_I)}{\partial\xi} = -2p(\xi_i - \xi_I)\left(d_{il}^2 + (\gamma\, d_a)\right)^{p-1} \tag{100}$$

3.2.2. Remaining Implementation

The rest of the implementation of the RPIM and the NNRPIM is very similar to the FEM method. Therefore, Section 3.1.4 can be followed to build the stiffness matrix in NNRPIM with the shape functions and its derivatives being substituted by the shape functions and derivatives from Section 3.2.1, and where previously deformation matrix $\boldsymbol{B}_I$ would be built based on the number of nodes inside the element, now it is based on the number of nodes inside the influence cell or influence domain of the interest point.

The natural boundary conditions can also be imposed following Section 3.1.5 but using the shape functions from Section 3.2.1 as explained before.

Since the RPI shape functions have the Kronecker delta property, the essential boundary conditions can also be imposed following Section 3.1.6. This is an advantage of using interpolant functions instead of approximant functions, such as the MLS, which do not possess this property and due to that must have special care when imposing these boundary conditions.

Finally, the stresses, strains and displacements can be calculated following Section 3.2.2.

Chapter 4

Computational Fracture Mechanics

4.1. Finite Element Method

The propagation of cracks is a widely studied problem in fracture mechanics and over the years there have been various approaches to the simulation of crack propagation and the stress field around the crack tip using FEM. Some of these approaches are: remeshing after each step (such as splitting elements at the crack tip or moving crack tip elements), nodal release at the crack tip, special crack tip elements, mostly used to obtain accurate stress field around the crack tip and strain and/or displacement enrichment, which constitutes the basis of the extended finite element method (XFEM).

The simplest way to simulate crack propagation is the nodal release technique, it simply "disconnects" a node, so the crack is increased by a Δa increment, along the element's edge until the next node, which now is the crack tip and the original node is split in two, therefore adding an additional node to the model, as shown in Figure 17.

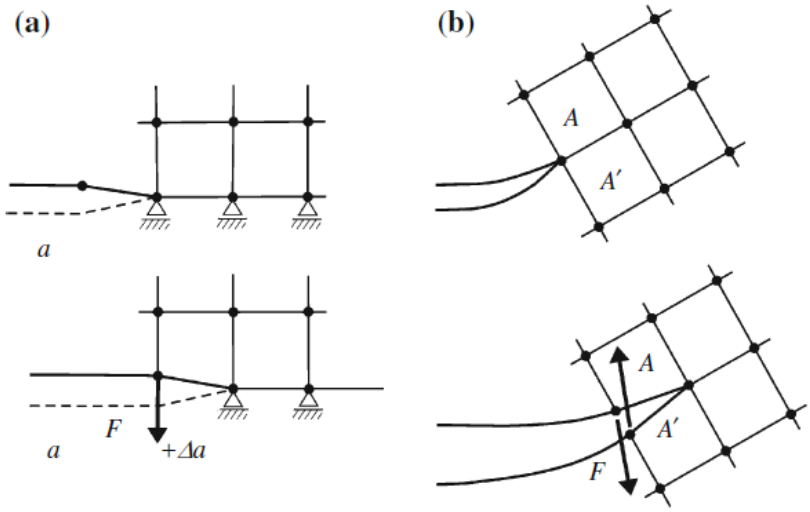

Figure 17 - Example of Node Release in (a) Mode I (b) Mixed Mode [6]

This technique is easy to implement and for a pure mode I loading it provides good results. Under mixed mode loading, which makes the crack propagation direction an unknown, this technique is limited by the mesh since the crack grows along the side of the element, making

this technique only suitable for problems where the crack path is known, and the user can arrange the mesh to fit the known path or when the mesh is very fine.

There are some authors that use this technique with some improvements, generally combined with some form of remeshing. For instance, Bouchard et al. [54] refined the mesh near the crack tip for each step of the propagation, to apply this technique to more general cases.

An interesting technique is the element splitting technique, this technique is more difficult to implement in 3D problems [55], due to the complexity of splitting 3D elements, but this study will focus on 2D problems where it is easier to implement. This technique has a clear advantage, the crack propagation is not affected by the mesh configuration. When the crack propagates, it simply splits the element along the crack, generating new elements, the obvious drawback here is that this technique can greatly increase the number of elements of the model, and it can also potentially lead to highly distorted elements. The splitting of the element can be done in many different ways and some examples can be visualised in Figure 18 for triangular elements.

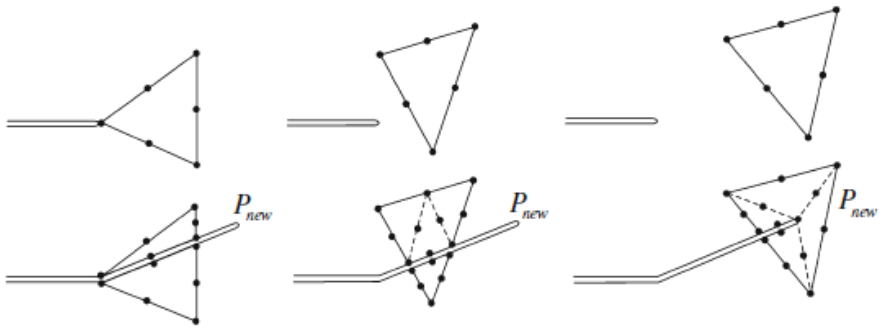

Figure 18 - Examples of element splitting [6]

Colombo in [56] developed a FEM code that splits the elements where the crack passes through when it propagates, in order to model the crack propagation. However, the SIF, the stress and displacement fields produced by this model were not very accurate. So, Colombo uses a sub model that uses quarter point elements to calculate the SIF, the stress and displacement fields. As proved in [57] quarter point elements provide solutions that accurately represent the singularity at the crack tip.

More recently, [58] introduced a method to model cracks using element splitting and mesh refinement, however this technique is only applied to model existing cracks, not to model crack propagation.

In [59] a technique is presented that does the refinement of triangular meshes in the vicinity of the crack and it can also perform its coarsening after the crack propagation in an area under evaluation. It is important to note that the coarsening criterion used in this paper requires a previous knowledge of the studied problem. Also, this technique, under mixed mode loading, presented some crack branching with the formation of microcracks along the main crack.

The main problem with these techniques that require some form of remeshing, either by cutting elements or refining the mesh, is the transferring of field variables between two different discretizations and to maintain a data structure to retrieve adjacency relationships efficiently [59]. However, this problem is not present when dealing with brittle materials where there is no plastic deformation.

The origin of XFEM can be traced back to 1999. Moës, Dolbow and Belytschko developed a method based on FEM with an enriched displacement field near the crack tip and also additional degrees of freedom [60]. The advantage of XFEM is that the crack is represented independently from the mesh, making the remeshing unnecessary. The drawback of this method is the variable number of degrees of freedom per node. This is a technique has gathered significant interest since its introduction and there are several available papers focused on it [61][62][63].

4.2. Meshless Methods

It is simpler to simulate crack propagation in meshless methods because they are not formulated with elements, but with nodes. This feature eliminates the main problem with FEM formulation which is the remeshing after each propagation step.

There are various different approaches to model discontinuities using meshless methods, in the following sections, some of the most common approaches are presented in this section.

4.2.1. Visibility Method

The first time that a meshless method was used to model cracks it used the visibility method [36], [64]. This method is good because of its simplicity. It considers that the crack boundary is opaque, this means that the nodes on the opposite side of the crack are not considered when defining the influence domain of an interest point, as Figure 19 shows. The grey areas are excluded from the influence domains of the interest point $\boldsymbol{x}_I$. However, this method leads to undesired discontinuous shape functions within the domain for interest points near the crack tip [3]. Nonetheless, the results obtained with this method seem to converge [65].

Figure 19 - Influence domains near the crack using the visibility method

4.2.2. Diffraction Method

The diffraction method was inspired by the way light diffracts around a sharp corner, it was introduced in [66] as an improvement to the visibility method. Using this method, the distance between the interest point and a node is defined by the following function:

$$d_{iI} = \left(\frac{S_1 + S_2(\boldsymbol{x}_i)}{S_0(\boldsymbol{x}_i)}\right)^{\lambda} S_0(\boldsymbol{x}_i) \tag{101}$$

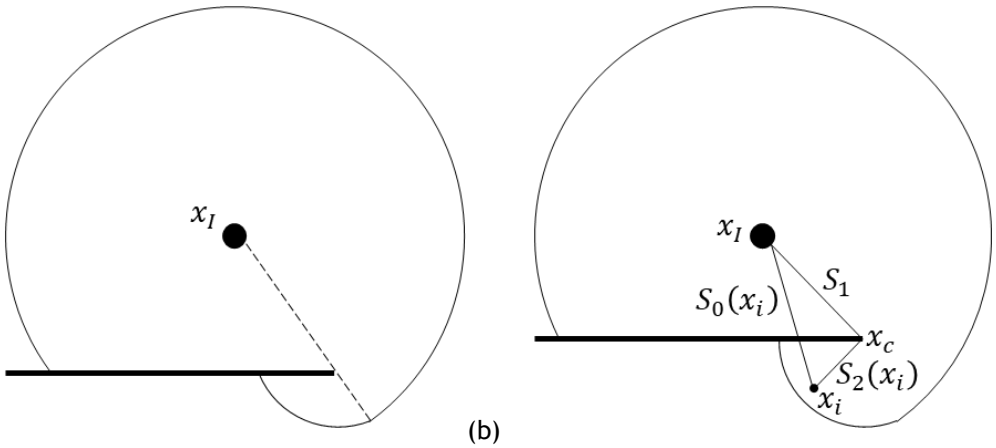

Figure 20 - (a) Influence domain near the crack tip using the diffraction method (b) Scheme of the calculation of d_{iI}

where $S_0(\boldsymbol{x})$ is the distance between the interest point and the node, S_1 is the distance between the interest point and the crack tip and $S_2(\boldsymbol{x})$ is the distance between the crack tip and the node, as seen in Figure 20b, these are mathematically defined as follows:

$$S_0(\boldsymbol{x}_i) = \|\boldsymbol{x}_i - \boldsymbol{x}_I\|, \qquad S_1(\boldsymbol{x}) = \|\boldsymbol{x}_c - \boldsymbol{x}_I\|, \qquad S_2(\boldsymbol{x}_i) = \|\boldsymbol{x}_i - \boldsymbol{x}_c\| \tag{102}$$

For nodes in the visible region d_{iI} remains the same because in that case $S_1 + S_2 = S_0$, λ is a parameter that defines the size of the influence domain behind the crack, $\lambda = 1 \vee \lambda = 2$ yields reasonable results [66]. Using this method, the resulting shape functions are continuous within the domain and discontinuous in the crack, as intended.

4.2.3. Transparency Method

The transparency method was also introduced in [66] as an alternative to the diffraction method. The idea behind this method is to make the crack transparent near the tip, being the degree of transparency related to the distance from the crack tip to the point of intersection.

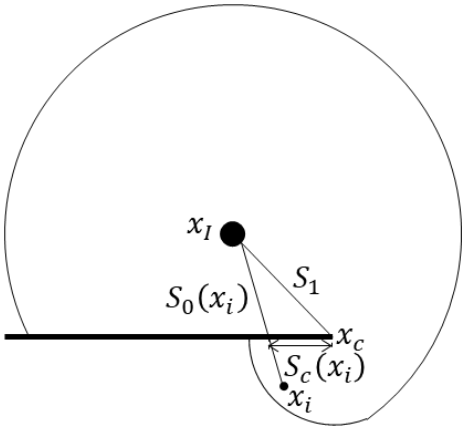

Figure 21 - Scheme of the calculation of d_{iI} using the transparency method

$$d_{iI}(\boldsymbol{x}_i) = S_0(\boldsymbol{x}_i) + d_{mI}\left(\frac{S_c(\boldsymbol{x}_i)}{\bar{S}_c}\right)^{\lambda}, \qquad \lambda \geq 2 \tag{103}$$

being d_{ml} the radius of the influence domain, $S_c(\boldsymbol{x}_i)$ the distance from the crack tip to the intersection point, as seen in Figure 21 and $\bar{S}_c$ is a parameter that sets the intersection distance where the crack is completely opaque, being defined by:

$$\bar{S}_c = \kappa h \tag{104}$$

Where κ is used to vary the opacity and h is the measure of nodal spacing. Like the diffraction method, this method is continuous within the domain and discontinuous across the crack.

4.2.4. Shape Function Enrichment

Another alternative to obtain the correct results using meshless methods in problems with cracks is the enrichment of the shape functions. This enrichment can be further divided into two different types, intrinsic enrichment and extrinsic enrichment [67], but these definitions may vary between different publications. In this paper the following definitions were used, intrinsic enrichment is modifying the shape functions without adding extra terms, while extrinsic enrichment is defined as adding extra terms to the moving least square (MLS) shape functions. Although this work does not use MLS shape functions (it uses RPI shape functions), the enrichment procedure will be presented for the MLS shape functions.

The first work to propose the shape function enrichment in meshless methods was [68] by Fleming et al.. This work presented an extrinsic enrichment of the MLS shape functions by adding the terms of the near-tip asymptotic expansion of the displacement field at the crack tip to the trial function as follows:

$$u_\alpha^h(\boldsymbol{x}) = p^T(\boldsymbol{x})a_\alpha(\boldsymbol{x}) + \sum_{j=1}^{n_c}[k_1^j Q_{1\alpha}^j(\boldsymbol{x}) + k_2^j Q_{2\alpha}^j(\boldsymbol{x})], \qquad \alpha = 1,2 \tag{105}$$

being $u_\alpha^h(\boldsymbol{x})$ the approximation of the exact solution $u_\alpha(\boldsymbol{x})$, $p(\boldsymbol{x})$ is the polynomial basis, which varies based on the Pascal triangle presented in Figure 9, n_c is the number of cracks in the problem, $a_\alpha(\boldsymbol{x})$ are the coefficients of the polynomials, α represents the Cartesian components and k_1^j and k_2^j are unknown constants associated with the crack j. The equations presented here are the ones from [68], which belong to the EFGM, but they could also be applied to the shape functions of the RPIM and NNRPIM.

The functions $Q_{1\alpha}(\boldsymbol{x})$ and $Q_{2\alpha}(\boldsymbol{x})$ describe the near-tip displacement field and are defined as:

$$Q_{11}(\boldsymbol{x}) = \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\cos\frac{\theta}{2}\left[\kappa - 1 + 2\sin^2\frac{\theta}{2}\right] \tag{106}$$

$$Q_{12}(\boldsymbol{x}) = \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\sin\frac{\theta}{2}\left[\kappa + 1 - 2\cos^2\frac{\theta}{2}\right] \tag{107}$$

$$Q_{21}(\boldsymbol{x}) = \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\sin\frac{\theta}{2}\left[\kappa + 1 + 2\cos^2\frac{\theta}{2}\right] \tag{108}$$

$$Q_{22}(\boldsymbol{x}) = -\frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\cos\frac{\theta}{2}\left[\kappa - 1 - 2\sin^2\frac{\theta}{2}\right] \tag{109}$$

where r and θ are the radial coordinates with origin at the crack tip, as described in Figure 6, μ is the shear modulus and κ is the Kolosov constant defined as:

$$\kappa = \begin{cases} 3 - 4\nu \\ (3 - \nu)/(1 + \nu) \end{cases} \tag{110}$$

After some transformations, presented in detail in [68], the trial function can be written as:

$$u_\alpha^h(\boldsymbol{x}) = \sum_{I=1}^{\mathrm{n}} \phi_I(\boldsymbol{x}) \tilde{u}_{I\alpha} + \sum_{j=1}^{n_c} \left[k_1^j Q_{1\alpha}^j(\boldsymbol{x}) + k_2^j Q_{2\alpha}^j(\boldsymbol{x}) \right] \tag{111}$$

being $\phi_I(\boldsymbol{x})$ the shape function and $\tilde{u}_{I\alpha}$ the modified nodal coefficients, defined as:

$$\phi_I(\boldsymbol{x}) = p^T(\boldsymbol{x}) A^{-1}(\boldsymbol{x}) P_I(\boldsymbol{x}) \tag{112}$$

$$\tilde{u}_{I\alpha} = \tilde{u}_{I\alpha} - \sum_{j=1}^{n_c} \left[k_1^j Q_{1\alpha}^j(\boldsymbol{x}) + k_2^j Q_{2\alpha}^j(\boldsymbol{x}) \right] \tag{113}$$

where $A(\boldsymbol{x})$ and $P_I(\boldsymbol{x})$ are:

$$A(\boldsymbol{x}) = \sum_{I=1}^{n} w(\boldsymbol{x} - \boldsymbol{x}_I) p(\boldsymbol{x}_I) p^T(\boldsymbol{x}_I) \tag{114}$$

$$P_I(\boldsymbol{x}) = [w(\boldsymbol{x} - \boldsymbol{x}_1) p(\boldsymbol{x}_1), \dots, w(\boldsymbol{x} - x_n) p(\boldsymbol{x}_n)] \tag{115}$$

In the same paper, Fleming et al. also introduced the intrinsic enrichment by expanding the polynomial basis to include the asymptotic field near the crack tip. After some manipulation using trigonometric identities, it can be shown that the functions from Equations 106 to 109 can be included in the basis, which when using a linear basis would resemble as:

$$\boldsymbol{p}^T(\boldsymbol{x}) = \left[1, x, y, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}\sin\theta, \sqrt{r}\cos\frac{\theta}{2}\sin\theta \right] \tag{116}$$

This enrichment can be performed to any polynomial basis, the linear basis was used in this example for its simplicity. The displacement field is given by:

$$u_\alpha^h(\boldsymbol{x}_I) = \sum_{\mathrm{I}=1}^{\mathrm{n}} \boldsymbol{p}^T(\boldsymbol{x}_I) \boldsymbol{A}^{-1}(\boldsymbol{x}) \boldsymbol{p}(\boldsymbol{x}_I) u_{I\alpha} \tag{117}$$

With the definition of shape function from Equation 112, this can be rewritten as:

$$u_\alpha^h(\boldsymbol{x}_I) = \sum_{\mathrm{I}=1}^{\mathrm{n}} \phi_I(\boldsymbol{x}_I) u_{I\alpha} \tag{118}$$

The main advantage of this enrichment compared to the extrinsic enrichment is that it does not involve any additional unknowns. However, due to the bigger size of the basis, the computational cost to invert the moment matrix $\boldsymbol{A}(\boldsymbol{x}_I)$ is increased. Another disadvantage of this approach is that for a problem with more than one crack, additional terms would have to be added to the basis for each crack.

Fleming et al. also noted that this type of enrichment can lead to ill-conditioned moment matrices and proposed some solutions to alleviate this problem [68] such as, diagonalizing the moment matrix by Gram-Schmidt orthogonalization.

Chapter 5

Crack Opening Path Algorithm

5.1. Algorithm Overview

The developed crack propagation algorithm can be used in FEM, RPIM and NNRPIM, being one of the goals of this work the comparison between these different numerical methods in crack propagation problems.

The first step is to discretize the domain in nodes and elements. Then, the integration points can be obtained.

When the FEM is used the discontinuity in the domain, the crack, is achieved through the elements, since elements on different sides of the crack do not interact with each other directly. In meshless methods, RPIM and NNRPIM, the discontinuity is simulated using the concept of groups. In essence, the group concept works as a visibility criterion, described in section 4.2.1. Therefore, the nodes on one side of the crack are part of one group and they do not see (connect with) the nodes on the other side of the crack, this is exemplified in Figure 22. Here, the nodes to the left of the crack will not be seen by the nodes on the right side of the crack and vice-versa, but they will both see the group above the crack.

It is now possible to obtain the influence domains for RPIM, knowing the groups of each node, as well as the natural neighbours in the NNRPIM.

Then, the usual process to obtain stress fields from numerical methods can be followed, first the interpolation functions are calculated, then the stiffness matrix is obtained. With the stiffness matrix calculated, the boundary conditions are applied. Finally, after the displacement vector is calculated, the stress and strain of each integration point can be calculated.

At this point, the crack propagation algorithm enters the routine to predict the direction of propagation and to increase the crack by the defined length. After the crack propagation is done for this iteration, it checks if this new length of the crack is the final crack length, defined by the user, if it is the stress and strain fields are calculated for this final crack length and then the it stops propagating; if it is not the final crack length the whole process is automatically repeated as many times as needed until the final crack length is reached.

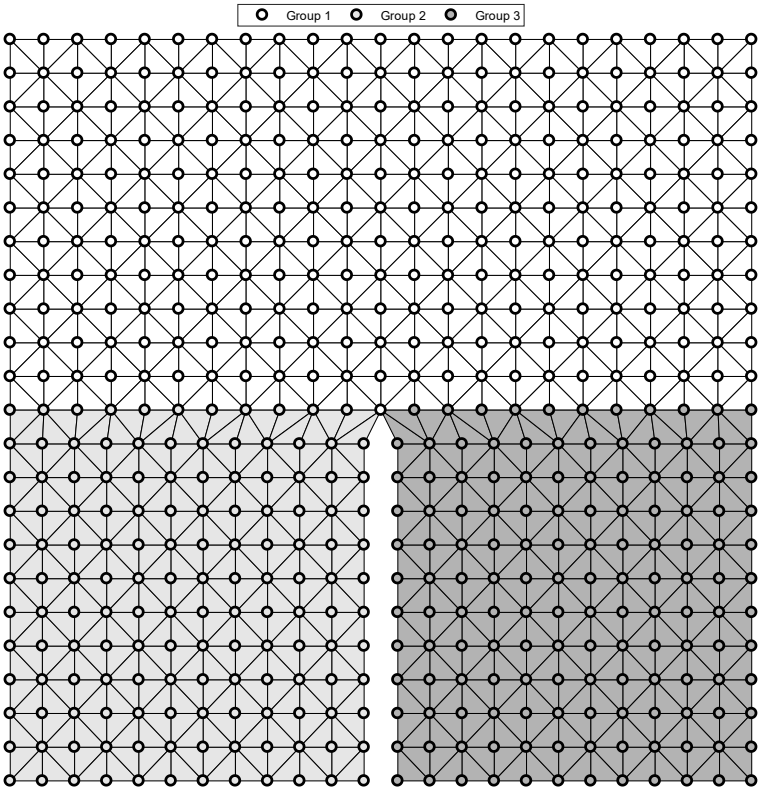

Figure 22 - Example of the groups for the elements and nodes at the beginning

5.1.1. Crack Propagation Algorithm

The direction of the crack propagation is determined using the MTS criterion referred in section 2.3.2. This criterion establishes that the crack will propagate in the direction perpendicular to the maximum tangential stress at the crack tip, which is the second principal direction.

This was the chosen criterion because it is very simple to implement, and it provides accurate results in brittle fracture problems, however, the algorithm could be easily adapted to use another criterion.

To have more accurate results, the principal stresses, used to obtain the propagation direction, were not calculated using the stresses of just the crack tip node. Instead they were obtained with a weighted average of all the integration points associated with the nodes, or elements in the FEM, around the crack tip.

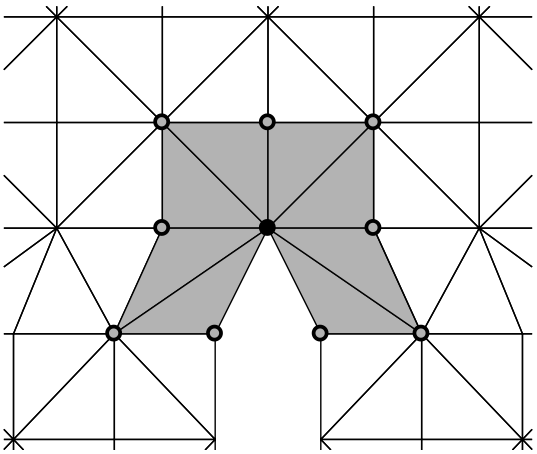

Figure 23 - Crack Tip and nodes and elements considered in the stress calculation for the propagation direction

Considering the crack tip shown in Figure 23, the stresses of the integration points of the grey elements would be the stresses considered in the calculation of the propagation direction when using the FEM, while using the meshless methods, the integration points associated with the grey nodes and the crack tip node, in black, would be the ones used to calculate the propagation direction.

Knowing the integration points that must be considered, the weighted stress at the crack tip is calculated using the following expression:

$$\sigma_{tip} = \frac{\sum_{I=1}^{n} \sigma_I w_I}{\sum_{I=1}^{n} w_I} \tag{119}$$

where n is the number of integration points near the crack tip and w_I is their weight in this calculation. The weight follows the expression:

$$w_I = \frac{2 d_{max}}{d_I} \tag{120}$$

being d_{max} the scalar distance between the crack tip and the node around the crack tip that is further away from it, which in Figure 23 would be the grey node at the bottom left corner, and d_I is the distance between the evaluated integration point I and the crack tip. This expression will ensure that when the integration point is very close to the crack its weight will tend to infinity, which is the behaviour of stress fields near the crack tip, and the furthest away it is from the crack tip, the lower its weight will be.

It is important to notice that the σ_{tip} from Equation 119 is only used to calculate the crack propagation direction, it does not overwrite the stress of the crack tip obtained previously.

Having σ_{tip}, it is possible to obtain its principal stresses and the respective principal directions following the process described in Section 2.1.2. The crack will then propagate along the second principal direction. The size of the propagation, Δa, in this algorithm is equal to the distance between the crack tip node and the node of the crack closest to the crack tip, excluding the crack tip itself, as seen in Figure 24a where the black dot is the old crack tip position and the white dot is the new crack tip position.

The crack tip node will move to its new position and two new nodes are created, which will be the new nodes of the crack closest to the tip, excluding the tip itself. These two new nodes will be perpendicular to the crack propagation direction and the distance between them is a fraction of the crack growth distance. An example of the new crack tip position and the two new nodes can be observed in Figure 24b, where the white dot is the new crack tip position and the grey squares represent the new crack nodes.

With the new crack tip position calculated, the next step is the pre-creation of the new elements and nodes surrounding the new crack tip position, this means that the attributes of these elements and nodes are created, but not yet inserted into the mesh.

Since the propagation direction is calculated using and average of the stresses around the crack tip, the more nodes and elements around the crack tip the better, to obtain smother results, but the elements also need to be undistorted. To ensure that there is always a reasonable number of element and nodes around the crack tip, and that the elements are undistorted this algorithm always has eight elements and nine nodes around the crack tip, disposed in a square manner.

To ensure that there are always eight elements surrounding the crack tip there are going to be created seven new nodes, in addition to the two nodes of the crack created in the previous step, disposed at the corners and middle points of a square, as seen in Figure 25, where the seven new nodes are the grey dots, and the future elements of the new crack tip position are in grey dashed lines.

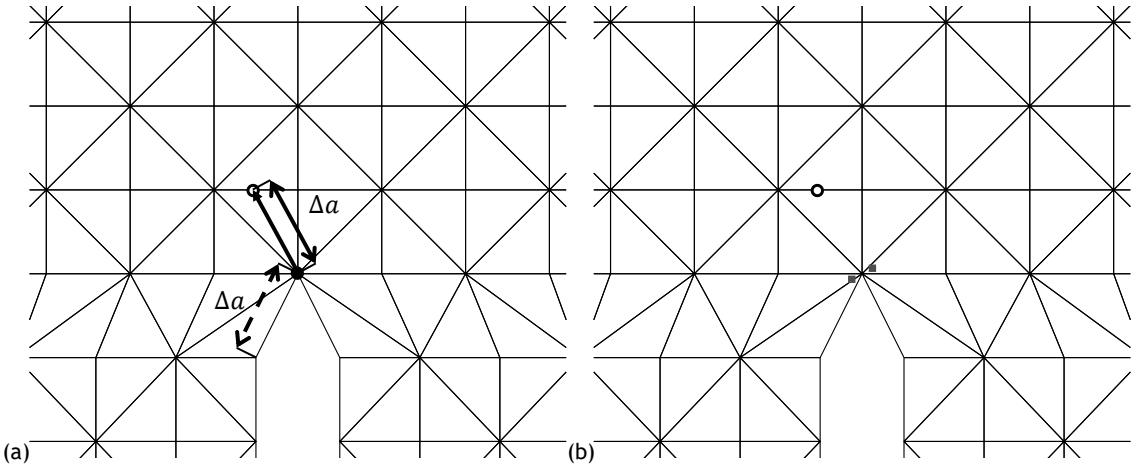

Figure 24 - (a) New crack tip position (b) and the two new nodes created from the previous crack tip position

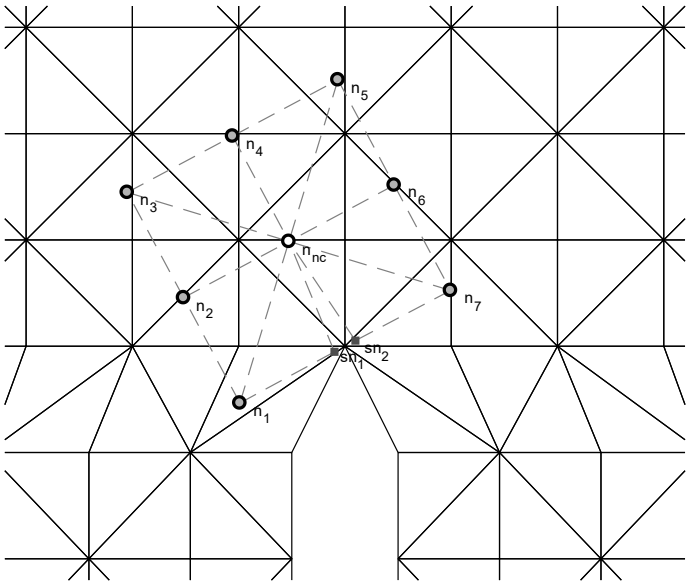

Figure 25 - New nodes and elements surrounding the crack tip

Knowing where the nodes and elements surrounding the new crack tip are going to be, the next step is to delete the nodes inside a circle centred at the crack tip, with radius equal to the distance between the new crack tip and one of the edges of the square formed by the elements surrounding the new crack tip, then a small tolerance, 10% of that distance, is added giving the final value for the radius, as seen in Figure 26.

There is one exception, when a node is inside the circle, but it is very close to either n_1 or n_7, in this case the node that was going to be deleted is merged with the n_1 or n_7 and the elements that had that node will now have n_1 or n_7. This usually does not occur in the first step of the crack propagation, but it happens almost always in every step of the crack propagation after it. An example of the merging is shown in Figure 27 where the light grey circles are the locations of the nodes that were merged with n_1 and n_7.

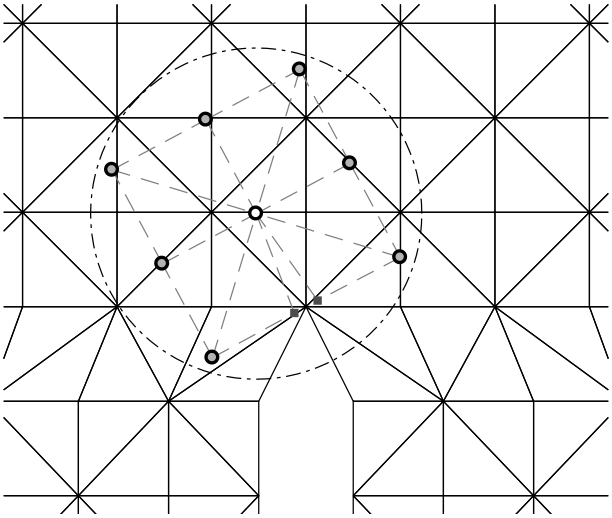

Figure 26 - Circle to determine the nodes that must be deleted

This merging happens almost always, unless there is a big change in the crack propagation direction from one iteration to the next, which usually should not occur. The merged nodes would be the nodes n_2 and n_6 of the previous iteration. A keen observer could also see that most times the nodes n_3 and n_5 of the previous iteration could also be merged with the nodes n_2 and n_6 of the new iteration, instead of deleted, but this is not done because it would sometimes cause element overlapping, so it was deemed a better option to just delete the nodes and elements connected to them and remesh after.

In this first iteration, there are no nodes merged with n_1 or n_7, so all the nodes inside the circle are deleted, the exception is the crack tip, and the nodes belonging to the crack boundary if they happen to be inside the circle, which is not the case here and it is very rare. The rule to delete elements is that they must have at least one node inside the circle, the only exception is if that node happens to be a merged node and they have no other nodes inside the circle. A new node is assigned to the elements that had the previous crack tip and have no other node inside the circle, to replace the previous crack tip, this node will be one of the two new nodes created when the crack tip was split, depending on which side of the crack the element is.

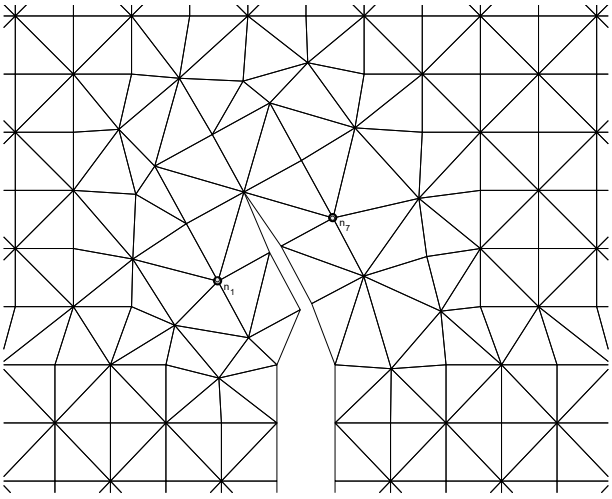

Figure 27 - Mesh after the second iteration

If node n_1 is not merged with another node, the elements that would have the split node from that side of the crack are also deleted, and the same happens when n_7 is not merged, but for the other side of the crack. This is done because without this some elements would become very distorted and with this extra rule those distorted elements would be less frequent.

There is an exception where no elements and nodes are deleted, and the nodes and elements of the square are not introduced into the mesh. This happens when one of the nodes inside the circle belongs to a boundary of the problem. Thus, to prevent deleting nodes in the boundary this whole process is not executed. In this case the crack propagates and the nearest node to the new crack position is merged with the crack tip node. If this node is a boundary node the new crack tip position coordinates are changed to that boundary node's coordinates and the analysis stops, if it is not a boundary node, the coordinates of the crack tip remain the same and the analysis continues to the next iteration.

Following these rules, the nodes and elements marked for deletion can be seen in Figure 28a, the nodes to delete are marked with an x and the elements to delete are greyed. Then, because nodes n_1 and n_7 were not merged with another node, the elements that were previously not marked and would have the split node respective to their side of the crack are also marked for deletion, as seen in Figure 28b.

The result, after deleting the nodes and elements, is shown in Figure 28c, the black nodes represent the nodes that must be connected by elements to the nodes of the square, called connector nodes. After, a small verification is performed to check if there are elements with three connector nodes, which there are in this case, shown in Figure 28d, these elements are then deleted and the connector node that belonged only to this element and no other is also deleted from the mesh and the connector node list. Without this, these elements would cause element overlapping in some rare cases. In the end, the resulting mesh around the new crack tip before the connector elements are created would look like Figure 28e.

The first step in the creation of these elements is to divide the black nodes into sectors, as shown in Figure 28f the sectors are defined by the dash-dotted lines and there are nine sectors in total. The point marked with a diamond used to divide sectors 1 and 9 is the point between the two nodes of the crack closest to the previous crack tip.

The nodes belonging to each sector are shown in Table 2, and the creation of the connector elements is done iteratively starting in sector one and ending in sector nine. Due to their similarities, sectors two, four, six and eight have the same rules for the creation of connector elements, with one exception for sectors two and eight because they are adjacent to sectors one and nine, respectively, which are unique sectors; then there is also another set of rules for sectors three, five and seven.

Table 2 - Nodes in each sector

Sector	Nodes Inside	Total Number of Nodes
1	cn_1	1
2	cn_2	1
3	cn_3	1
4	cn_4	1
5	cn_5, cn_6	2
6	cn_7	1
7	cn_8, cn_9	2
8	cn_{10}, cn_{11}	2
9	cn_{12}, cn_{13}	2

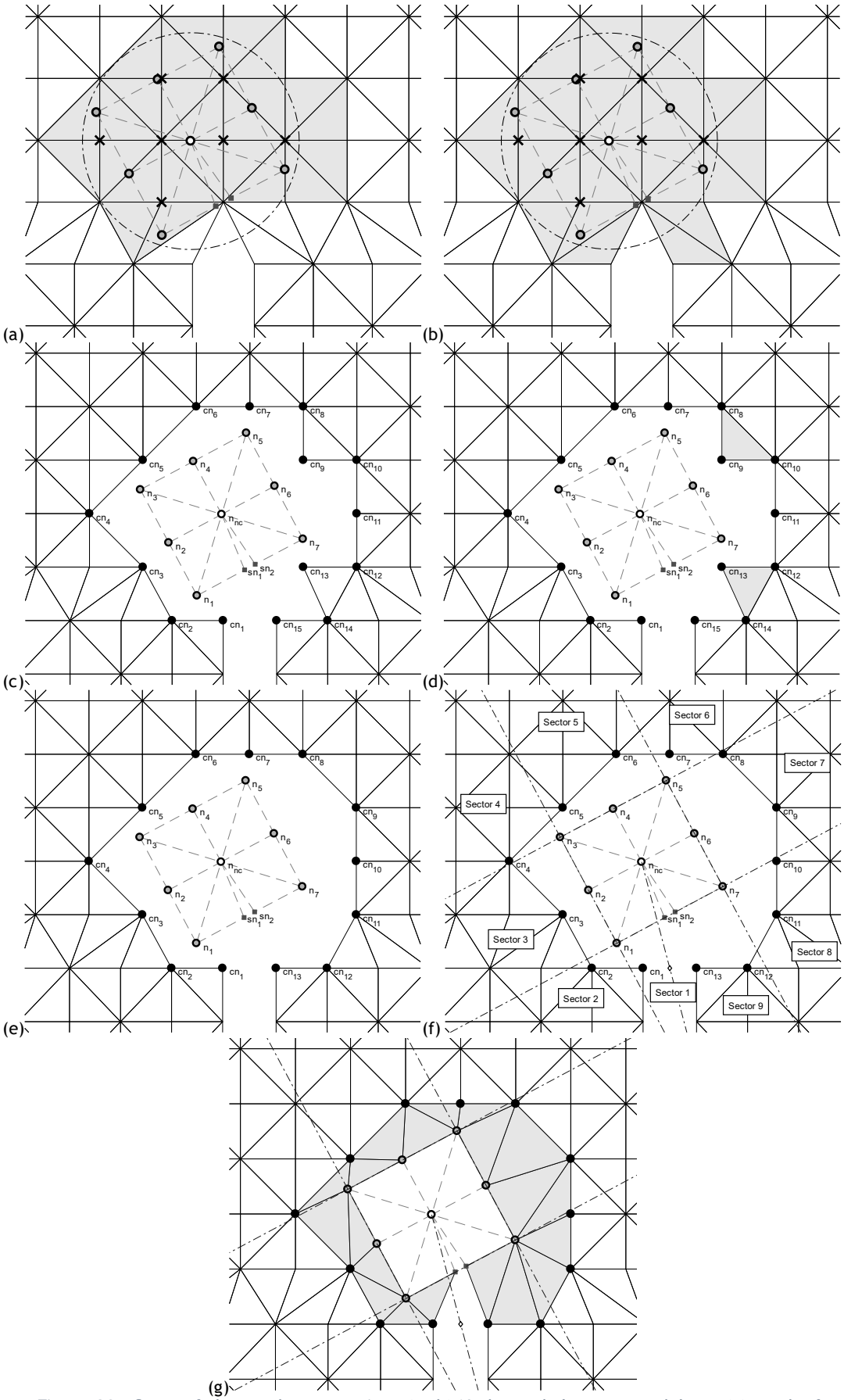

Figure 28 – Steps of the crack propagation, (a, b) Nodes and elements to delete, (c) mesh after they are deleted, (d) elements marked by the extra verification, (e) mesh before new elements are created, (f) sectors (g) and the elements connecting the crack tip elements to the rest

Sector one: if there are no connector nodes in the sector no new elements are created; if there is one connector node in this sector, one element is created with that connector node, n_1 and sn_1; if there are more that one connector node, a very rare case, the first connector node is created in the same manner and then subsequent connector elements that are created will have as their nodes each pair of two adjacent connector nodes and n_1. This means that the total number of connector elements created in this sector is equal to the number of connector nodes in the sector.

Sectors two, four, six and eight: these sectors serve as a link to the sectors adjacent to them. Because of this, if there are no connector nodes in the sector, one element is created having as its nodes the last connector node of the previous sector, the first connector node of the next sector and the square corner node respective to the sector, which is n_1 for sector one, n_3 for sector four, n_5 for sector six and n_7 for sector eight, this is a somewhat rare case; if there is one connector node in this sector, there are two new elements created, the first one will have as its nodes the last connector node from the previous sector, the first connector node of this sector and the square corner node (the exception here occurs for sector two when sector one has no connector nodes, in this case the node that was supposed to be the last connector node of the previous sector will be sn_1 instead) and the last element will have as its nodes the first connector node of the next sector, the last connector node of this sector and the square corner node (the exception here occurs for sector eight when sector nine has no connector nodes, in this case the node that was supposed to be the first connector node of the next sector will be sn_2 instead); if there are two or more connector nodes, the first and last elements will correspond to the first and last elements of the one connector node case, with the exceptions applied here too, but there will be a number of elements between those two that will have as their nodes each pair of two adjacent connector nodes and the corner node. This means that the total number of connector elements created in this sector is equal to the number of connector nodes in the sector plus one.

Sectors three, five and seven: In these sectors there should be at least one connector node, but in some very rare cases these sectors do not have any connector nodes. If that happens, only two elements are created, the first one will have as its nodes the last connector node of the previous sector and the first and the second square nodes of this sector, and the second element will have the first node of the next sector and the second and third square nodes of this sector; if there is one connector node in this sector, there are two new elements created, the first one will have as its nodes the first connector node of this sector, the first square node of the sector and the second square node of the sector and the last element will have as its nodes the last connector node of the sector, the second square node of the sector and the third square node of this sector; if there are two or more connector nodes, the first and last elements will correspond to the first and last elements of the one connector node case, but there will be a number of elements between those two that will have as their nodes each pair of two adjacent connector nodes and the second square node of the sector. This means that the total number of connector elements created in this sector is equal to the number of connector nodes in the sector plus one.

Sector nine: if there are no connector nodes in the sector, no new elements are created; if there is one connector node in this sector, one element is created with that connector node, n_7 and sn_2; if there are more that one connector node, a very rare case seen in Figure 28f, the first connector node is created in the same manner and then subsequent connector elements that are created will have has their nodes each pair of two adjacent connector nodes and n_7.

This means that the total number of connector elements created in this sector is equal to the number of connector nodes in the sector.

Following these rules, the resulting mesh after this process would look like Figure 28g where the grey elements are the newly created elements with this process and these rules.

Lastly, there is one final step to complete the remeshing part of the crack propagation process. This step consists in taking each connector node, the black dots in Figure 29, iteratively, search the nodes that are adjacent to it, i.e. nodes that share an element with the connector node, and then do an average of the coordinates of those nodes and change the coordinate of the connector node to that average (effectively centring the connector node between the nodes around it). The results of this process can be observed in Figure 29, this process was added because it results in less distorted connector elements. This step is not performed if one of the connecting nodes happens to be a boundary node, to prevent changing boundaries nodes from their original position.

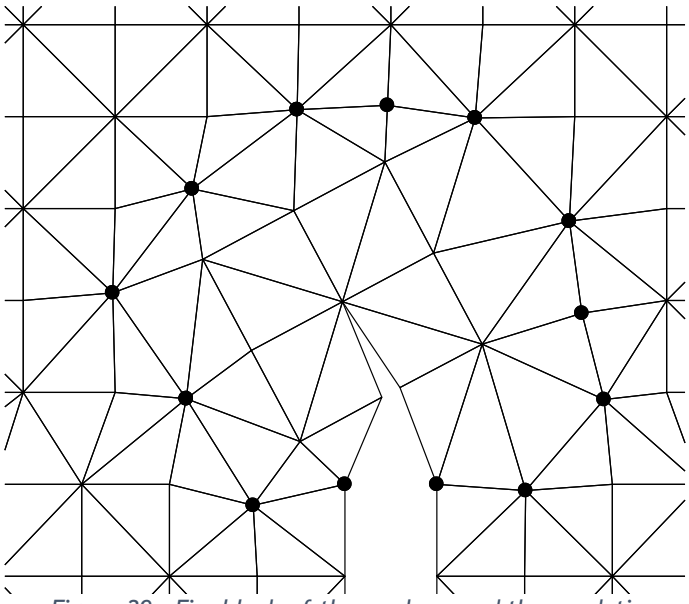

Figure 29 - Final look of the mesh around the crack tip

With the remeshing process concluded, it is time to redefine the groups of the nodes and elements, this is a relatively straightforward process when compared to the remeshing process. There are two planes created, as shown in Figure 30, plane one is perpendicular to the crack propagation direction and passes through the crack tip, this plane ensures that only nodes and elements below it can change to one of the groups of the crack sides. Plane two was created to divide the two sides of the crack, in this case the left side would be group two, and the right side would be group three. As a final auxiliary, a circle was created with its centre at the crack tip and radius equal to the triple of the distance between the new crack tip and one of the edges of the square formed by the elements surrounding the new crack tip, this circle limits the nodes and elements that have their group changed to the nodes and element that are inside it. For an element to be considered inside the circle, it must have all its nodes inside the circle, the same applies for knowing if the element is below or above plane one or to the right or left of plane two.

This means that nodes and elements that are below plane one, to the left of plane two and inside the circle will be changed to group two, unless they were already part of the group on the other side of the crack, in this case group three, in that instance their groups are not

changed because it is known that they are on the other side of the crack even though they are to the left of plane two. The same rules apply when changing the groups of elements and nodes to the right of plane two, but, instead of changing them to group two, they are changed to group three.

The groups are only changed inside the circle because the influence domains, or influence cells, in meshless methods act in a limited area around the nodes so there is no need to do this change for the whole domain, only for the nodes near the crack tip. Another reason is related to problems with multiple crack, where changing the groups for the whole domain could cause some problems in the visibilities around the crack tips, so this solution to limit the changes to a radius around the crack tip was devised.

These steps are repeated as many times as necessary until the final desired crack length is reached or until the crack propagates to a boundary node.

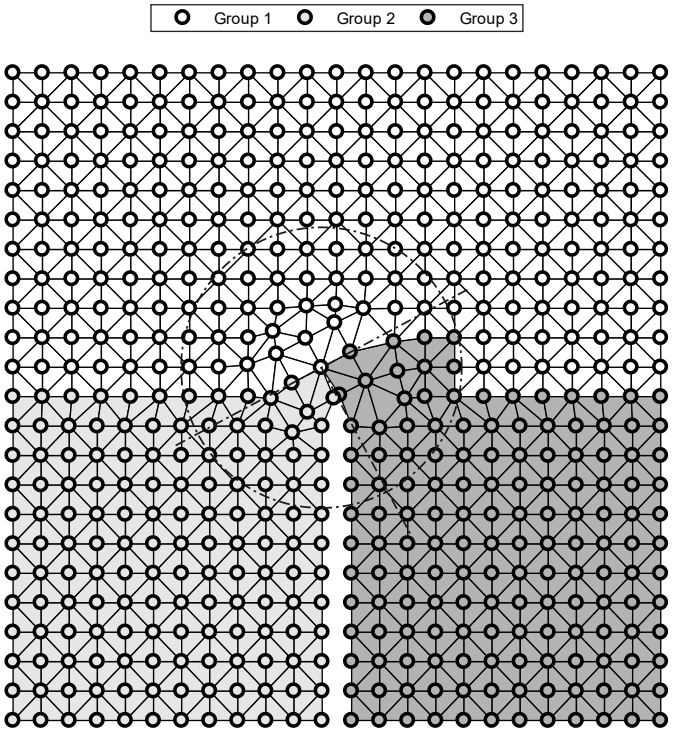

Figure 30 - Groups after the crack propagation step

Chapter 6

Numerical applications

6.1. Meshless initial parameters

As mentioned in Section 3.2, there are several parameters that have to be defined when using meshless methods in order to construct the shape functions and create the integration points. To maintain consistency these parameters remained the same for all the different benchmark tests.

So, according to the findings of Section 3.2, for RPIM the number of nodes inside each influence domain was set to 16, γ and p were defined as 1.03 and 1.42, respectively, the Gaussian quadrature, used for the creation of integration points, was the 1st quadrature. With this algorithm this means that each triangle of the background mesh is divided in three quadrilaterals and an integration point is created at the centre of each quadrilateral. Finally, a linear polynomial base was chosen.

For NNRPIM, a first order influence cell was chosen, as it was found to be better than the second order influence cell, in fracture problems, by Azevedo [4], γ and p were defined as 0.9999 and 0.0001, respectively, with first quadrature for the creation of integration points, which means one integration point for each division of a sub-cell, with a constant polynomial base and full integration.

6.2. Mode I Loading

The first example to be analysed with the proposed algorithm is the classic fracture mechanics example of an infinite plate under mode I loading. This is an interesting example to ascertain the quality of the crack propagation algorithm, because as it is known, under Mode I loading the crack is supposed to propagate forward without moving up or down, because the principal directions coincide with the Cartesian directions.

The scheme of the model used in this example is seen in Figure 31, since it is impossible to model an infinite plate, the dimensions used here are the ones shown in Figure 31. It was also possible to further simplify the example by identifying the symmetry along the $x = 1$ plane, so only half the model has to be used in the simulation. Additionally, a constrain at point $P(2,1)$ was added, in the middle of the right boundary, restricting the movement there

along the y axis only. This was assumed to ensure that both Cartesian directions have at least one degree of freedom constrained. Notice that, since that node is also not supposed to move along the y axis originally, this adaptation will not affect the results.

The mesh considered is not very fine, because this is a simple example. The mesh possesses 233 nodes and 400 elements initially, and it is shown in Figure 32. The material properties used were a Young's modulus of $1000\ Pa$ and a Poisson's ratio of 0.3. The loads applied at the top and at the bottom of the plate were equal to $1\ Pa$. The crack was set to stop after growing $0.8\ m$.

As expected, even using a coarse mesh, in this simple example the propagation occurs exactly as predicted, as seen in Figure 33. Independently of the method used the crack grew along the $y = 1$ plane, without going above or under it by a visible margin. Considering that the results obtained were already acceptable, there was no need to perform further tests using more refined meshes.

All the numerical methods took nine iterations to do the full crack propagation, and at the end they also had the same number of nodes and elements. At the end of the propagation the number of nodes increased to 241, this is due to the split nodes when the crack propagates, and the number of elements remained the same.

The resulting mesh after the crack propagation stops is shown in Figure 34. Because this is a simple example and the crack path was almost exactly the same for all the numerical methods, the resulting mesh at the end of the propagation was also equal for all the numerical methods. In this mesh there are no distorted elements and they all have very good proportions, the groups were also attributed correctly.

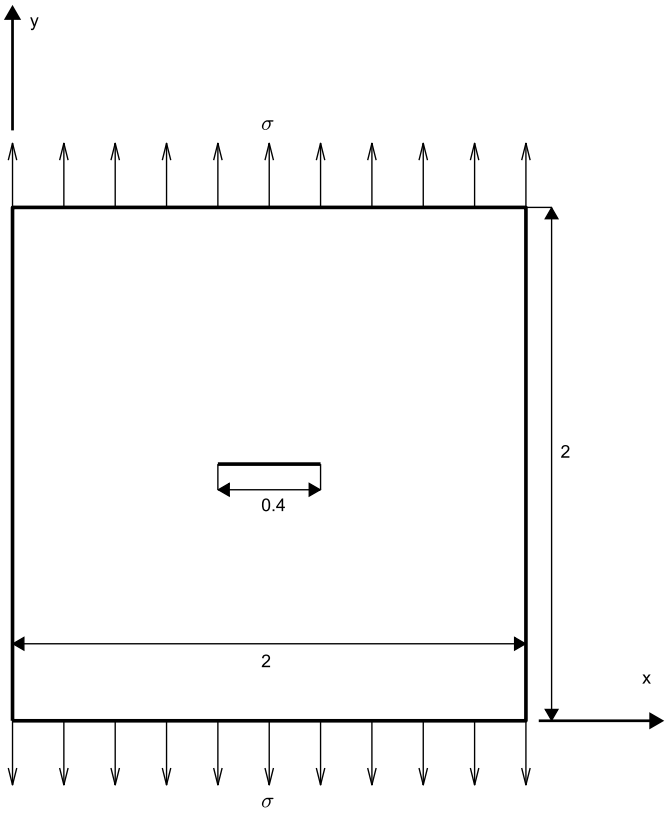

Figure 31 - Scheme of the Mode I loading example [measurements in m]

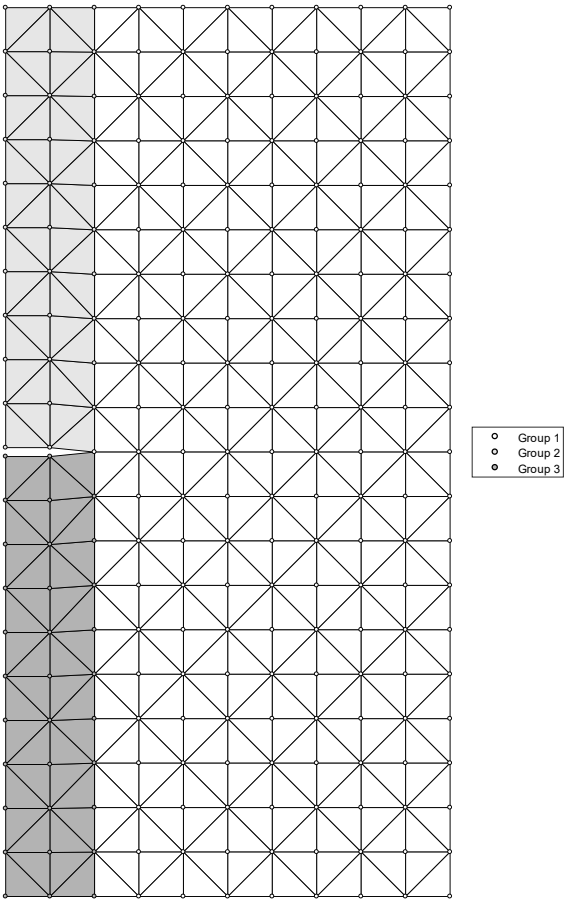

Figure 32 - Mode I 233 nodes mesh

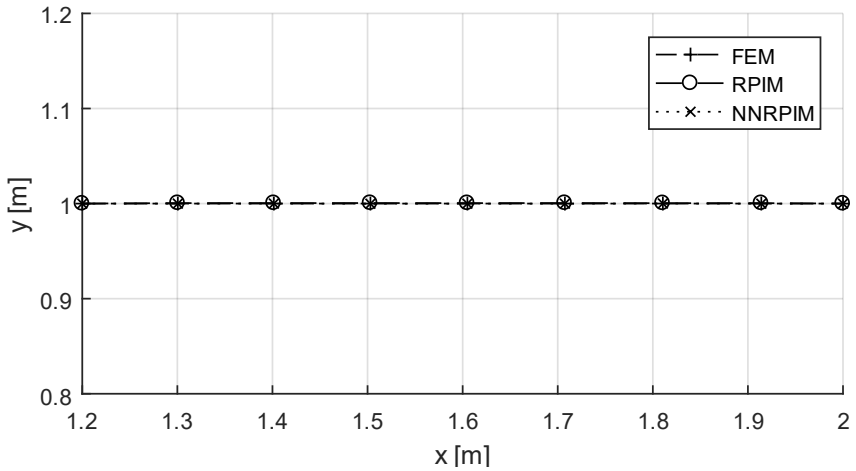

Figure 33 - Crack propagation under Mode I Loading

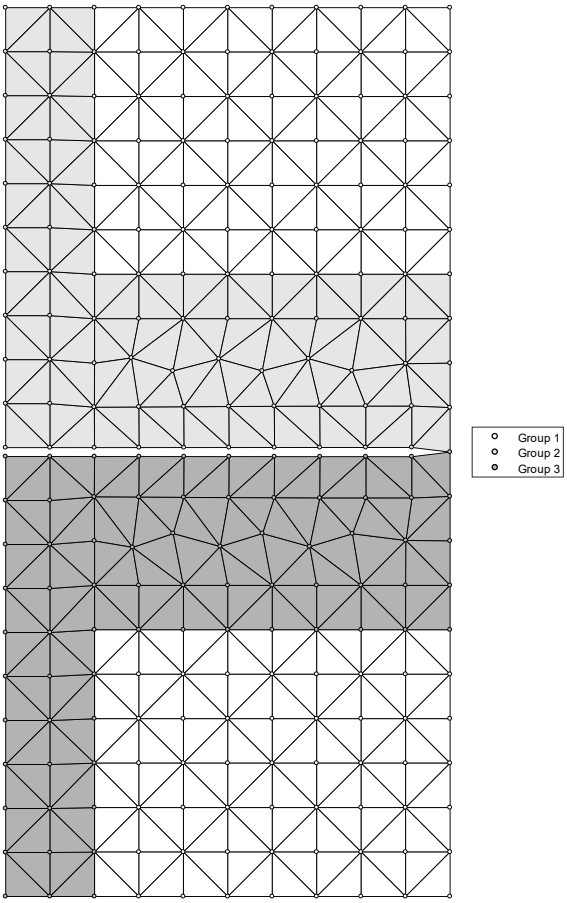

Figure 34 - Final mesh, FEM, RPIM and NNRPIM

6.3. Mode II Shear Loading

As a second test for the algorithm a more complex problem was chosen, the scheme of this problem is presented in Figure 35. In this example shear stress occurs, so the crack is not going to grow in a straight line. There is no theoretical result for the crack propagation of this example. Therefore, the results obtained here were compared to results from a previous work that used NNRPIM [4]. This example was also solved numerically by other past papers such as [69].

The material properties used were: Young's modulus of $1000\ Pa$ and a Poisson's ratio of 0.3. The load applied at the top of the plate was equal to $0.1\ Pa$ and as shown in the scheme, the movement of the plate was restricted in the x and y directions at its bottom.

Since this problem is more complex when compared to the previous one, the mesh was refined around the crack area for a second test to do a comparison of the results.

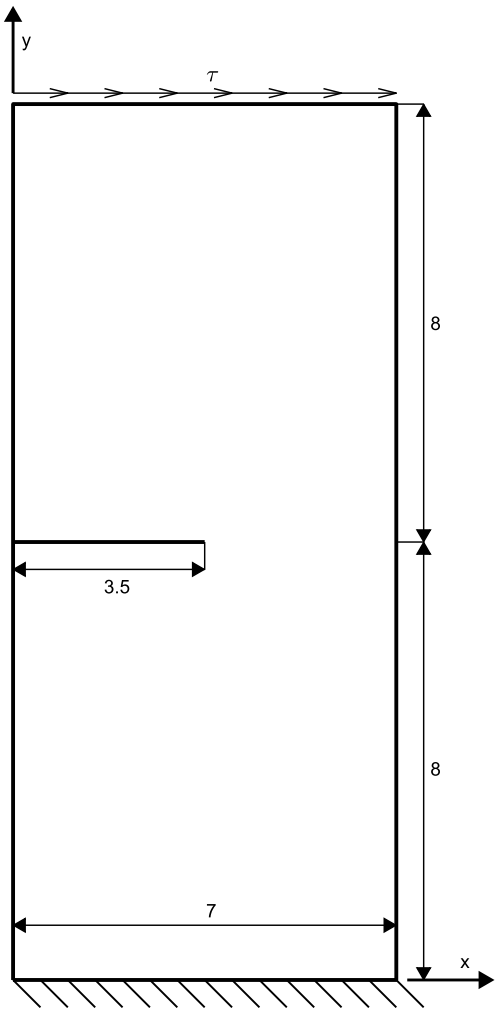

Figure 35 - Scheme of the Mode II shear loading example [measurements in m]

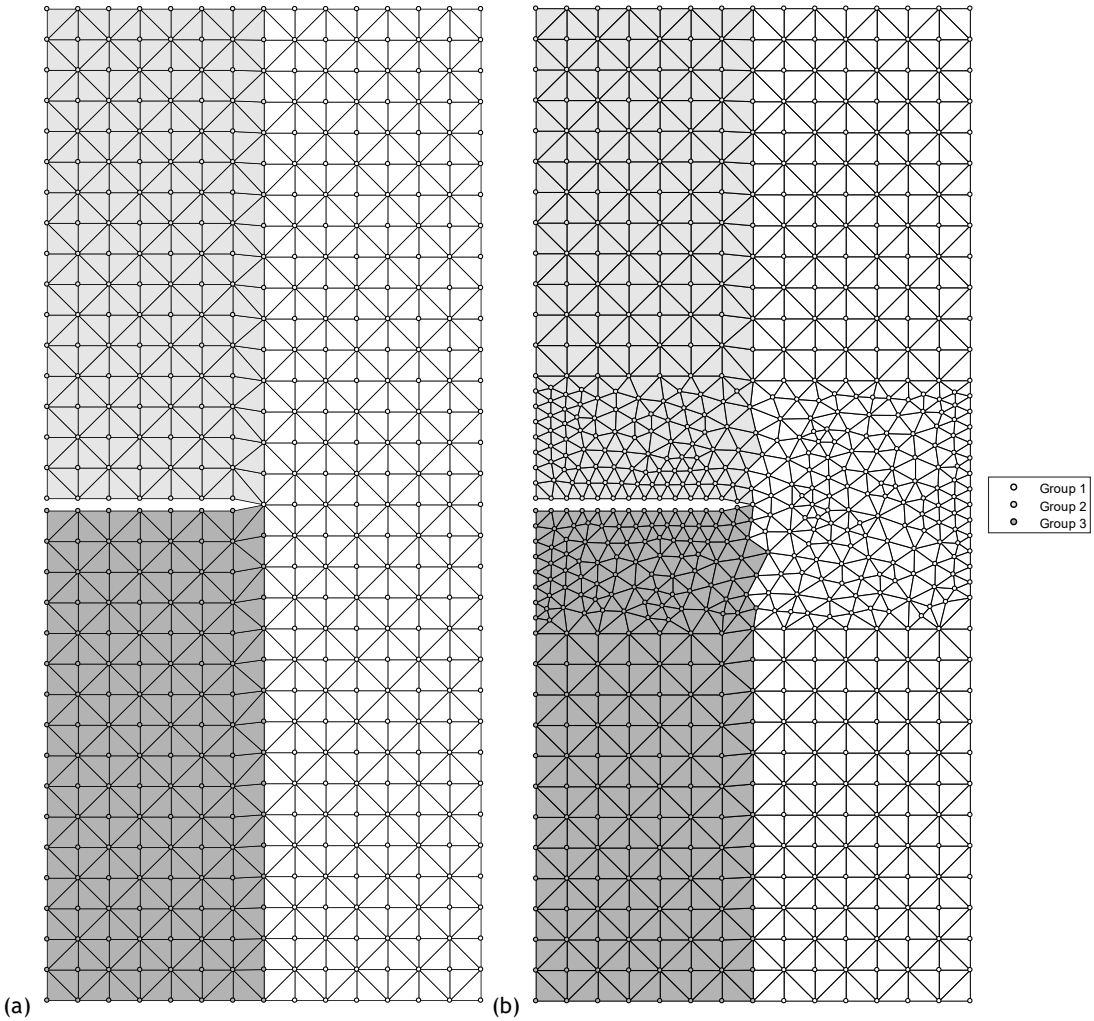

Figure 36 - Mode II shear (a) 502 nodes mesh (b) 725 nodes refined mesh

The first test was performed using the mesh presented in Figure 36a and its results are shown in Figure 37 which demonstrates that the mesh might have been too coarse to have an accurate crack path prediction. This mesh had initially 502 nodes and 896 elements.

In this test, all three numerical methods diverge from the results found in literature. When the crack grows beyond $x = 6$ the crack grows erratically, this is because this area has high stresses due to it being a very small strip holding the two halves of the domain together. The only exception to this is RPIM, in which the crack can grow until almost $x = 6.5$ before the erratic behaviour occurs. In such a sensitive area the stress fields cannot be accurately measured with a mesh this coarse and when the Δa is also big, due to being tied to the mesh size, it leads to big variances. The crack propagation was set to stop after the crack grew $3.5\ m$.

In the final step, using the FEM the number of nodes increased to 509 and the number of elements remained the same at 896. Using the RPIM the number of nodes increased to 508 and the number of elements decreased to 894. Using the NNRPIM the number of nodes increased to 509 and the number of elements remained the same at 896. Unlike in the Mode I example the number of nodes and elements changes from method to method because there are some slight differences in the propagation direction between them. With this mesh eight iterations were needed to reach the final crack length. Interestingly, the total number of elements actually decreased at the end of the crack propagation, when using RPIM, this is a very rare occurrence, but it can happen, especially with coarse meshes.

A state of the mesh in the iteration before the crack propagation changes directions abruptly is seen in Figure 38, for each of the numerical methods used. For FEM and NNRPIM that is the mesh at the third to last iteration and for RPIM that is the mesh at second to last iteration. Figure 38 shows that this algorithm, for this example and with this mesh, deals well with the remeshing necessary for the crack propagation and almost all the elements have a good shape and are not distorted. The only exception occurs in NNRPIM where there are two elements, at the top right corner of the elements surrounding the crack, which are somewhat distorted. This distortion should not have an effect here, since in NNRPIM element data is not used, because it is a meshless method. Nevertheless, it is important to notice these distortions, so they can be improved in case they would happen when using FEM.

In terms of groups, it is noticeable that the groups of elements and nodes in front of the crack are not changed, and the groups on each side of the crack were attributed correctly. There are some elements that have all their nodes belonging to a group while they are part of a different group. This is probably because some of their nodes changed groups because a neighbouring element was inside the circle in a previous iteration, and in an iteration further down the line the other nodes changed groups due to the same reason, but the nodes that were previously changed are not inside the circle of group changing in this iteration, so the group of that element is not changed. This shouldn't have an effect, because for meshless methods element data, is only used in RPIM in the creation of integration points.

As a final note, it is possible to observe the special case when connector nodes are in the boundary of the crack and these nodes are not moved. In the RPIM mesh, the special case when one of the nodes inside the circle of deletion is a boundary node is also observable. In this special case, crack simply propagates and merges with the nearest node, not doing the square process.

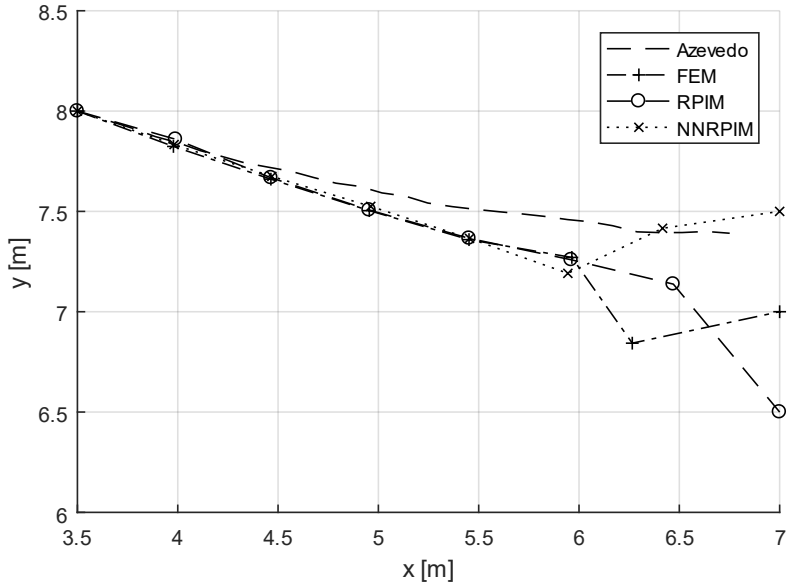

Figure 37 - Crack propagation under Mode II shear loading with 502 nodes mesh

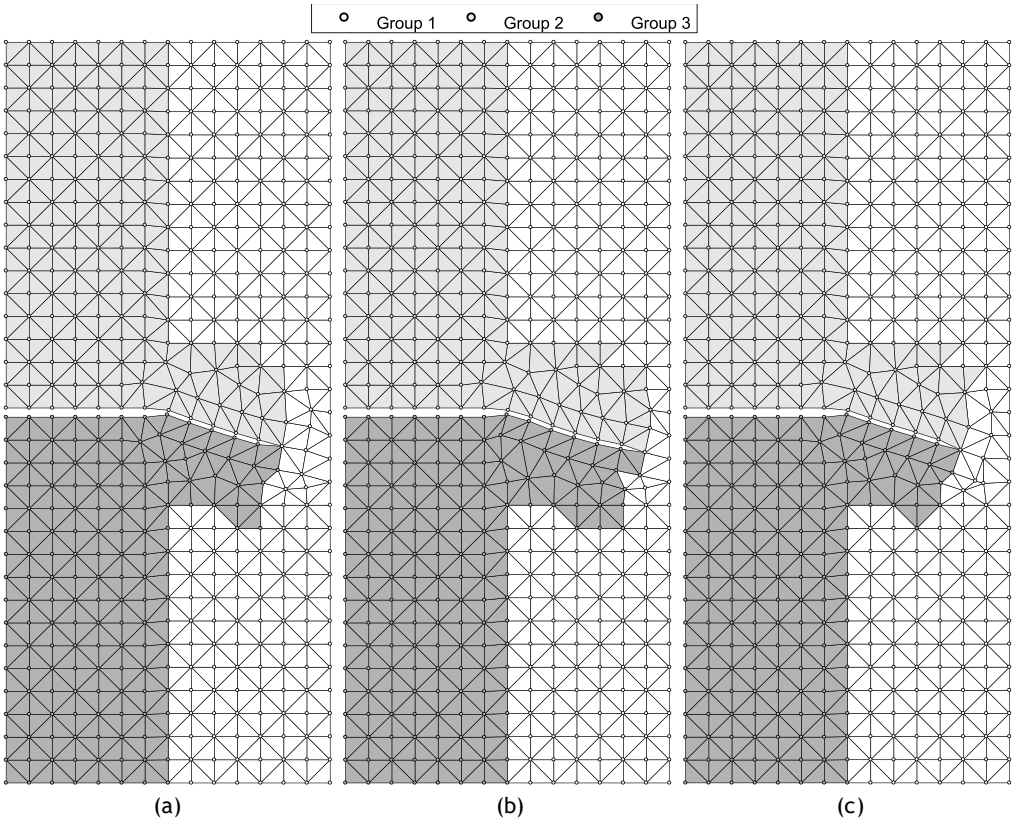

Figure 38 – Mode II shear final mesh (a) FEM, (b) RPIM, (c) NNRPIM with 502 nodes initially

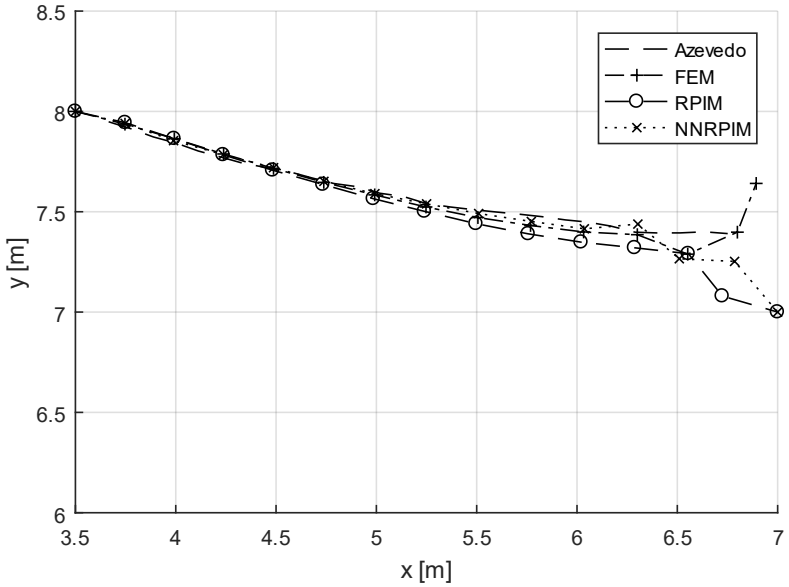

Figure 39 - Crack propagation under Mode II shear loading with 725 nodes refined mesh

Due to the less accurate results using the coarse 502 nodes mesh, it was necessary to perform a refinement in the crack tip area, the mesh after this refinement is seen in Figure 36b. This also serves as an opportunity to assess how the algorithm deals with more unstructured meshes. The new mesh had initially 725 nodes and 1313 elements, it represents an additional 233 nodes and 417 elements when compared to the unrefined mesh. The crack propagation was set to stop after the crack grew $3.5\,m$, the same as for the unrefined mesh.

The results, presented in Figure 39, show that with this refinement the crack path is very similar to the crack path achieved by Azevedo. There is some strange behaviour in the last three steps of FEM and NNRPIM and in the last two steps of RPIM, as in the previous case, but since the mesh is finer and Δa are smaller the crack could grow further than with the unrefined mesh without this erratic behaviour.

The method that managed to be closest to the results from previous papers seems to be NNRPIM, and RPIM seems to be the one furthest from the results of past papers, but even for RPIM the differences are minimal.

In the final step, using the FEM the number of nodes increased to 754 and the number of elements increased to 1343. Using the RPIM the number of nodes increased to 751 and the number of elements increased to 1337. Using the NNRPIM the number of nodes increased to 755 and the number of elements increased to 1345.

The total number of iterations to complete the propagation was 15 which is almost double the number of increments with the unrefined mesh, this was to be expected because Δa is approximately half of the Δa with the unrefined mesh, because it is tied with the distance between the crack tip and the node of the crack closest to it.

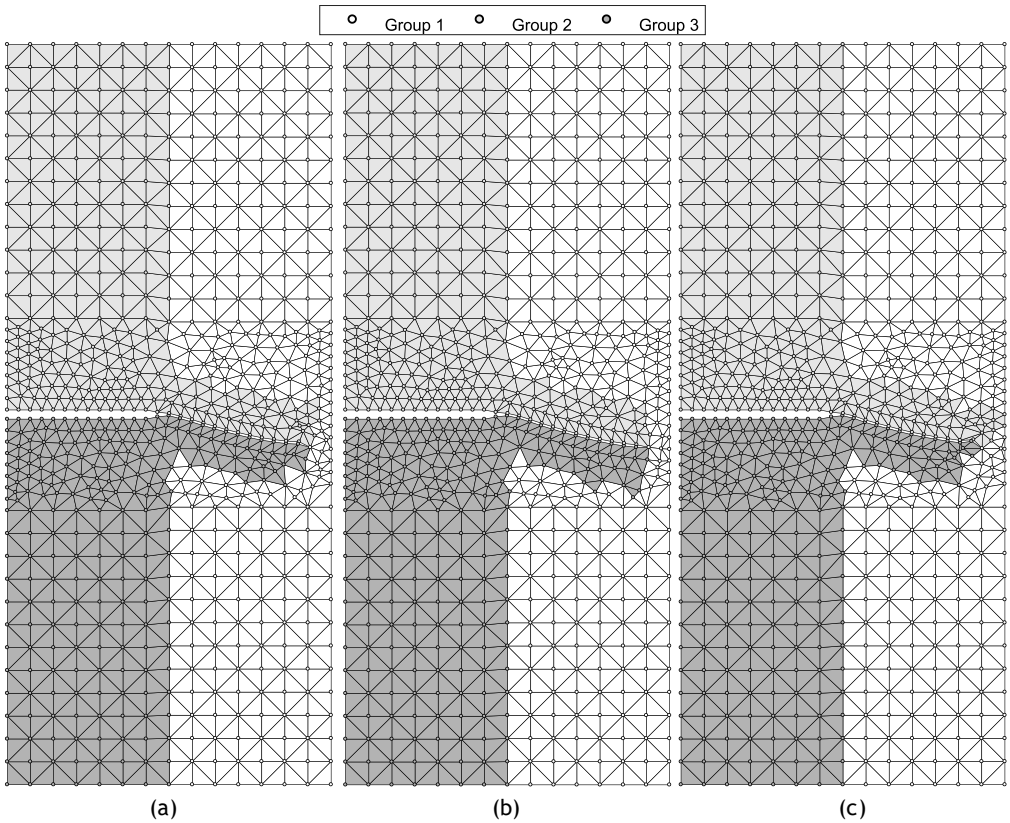

Figure 40 - Mode II shear final mesh (a) FEM, (b) RPIM, (c) NNRPIM with 725 nodes initially

A state of the mesh in the iteration before the crack propagation changes directions abruptly is seen in Figure 40, for each of the numerical methods used. In this case this is mesh at the third to last iteration for all the numerical methods. Figure 40 shows that this algorithm also deals well with the remeshing necessary for the crack propagation with this more refined and more irregular mesh, and almost all the elements have a good shape and are not distorted. The mesh in the RPIM case looks particularly good with elements with very good shapes. However, in NNRPIM there are some distorted elements, as in the unrefined mesh case. Again,

distortion should not have an effect here, but it is important to notice these distortions in case they would happen when using FEM.

In terms of groups they were also attributed correctly with this more refined mesh but there are also some cases where an element has all its nodes belonging to a group, while the element belongs to a different group.

6.4. Three Point Bending of a Beam

As the third test for this algorithm another problem was chosen, which consists of a beam simply supported in two opposite points and with a point load applied at its centre in the top boundary, the scheme with the dimensions and the boundary conditions of this problem is presented in Figure 41. The force applied to the beam was $1\ N$, the Young's modulus and the Poisson's ratio of the material are $1000\ MPa$ and 0.3 respectively. The crack propagation was set to stop after the crack grew $50\ mm$.

This problem was already studied in other research works, such as [4], [70], as a reference for the propagation, results obtained in [4] were used in this work as comparison solutions. As in the previous example, this problem was also studied using a normal mesh and then a mesh refined in the crack area. The initial mesh, before the crack propagates further is seen in Figure 42a and the refined mesh is in Figure 42b.

The crack propagation path, when using the unrefined mesh is shown in Figure 43, this mesh had initially 795 nodes and 1456 elements. In Figure 43, a different scale for each axis was used so the differences in the crack path between each method are better shown.

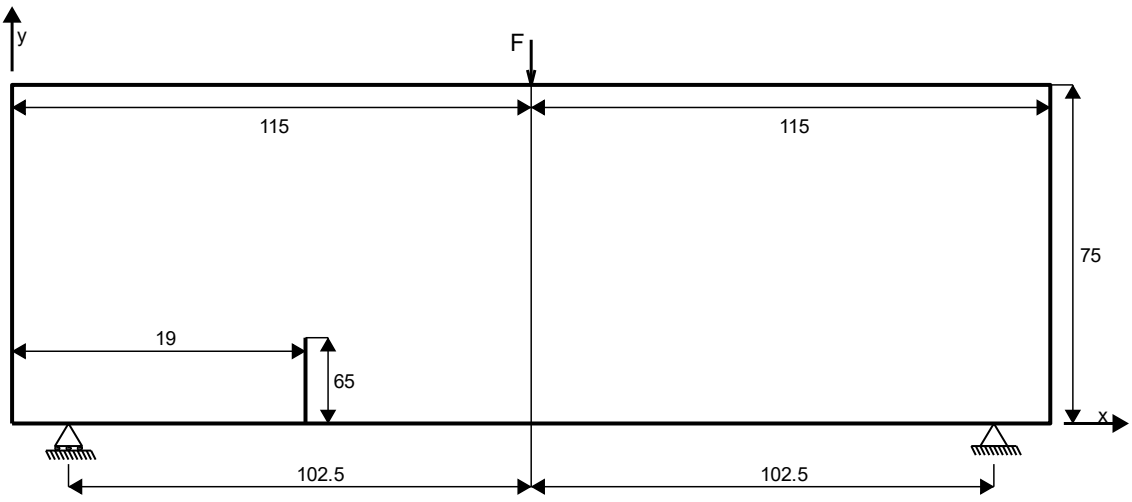

Figure 41 - Scheme of the three point bending of a beam example [measurements in mm]

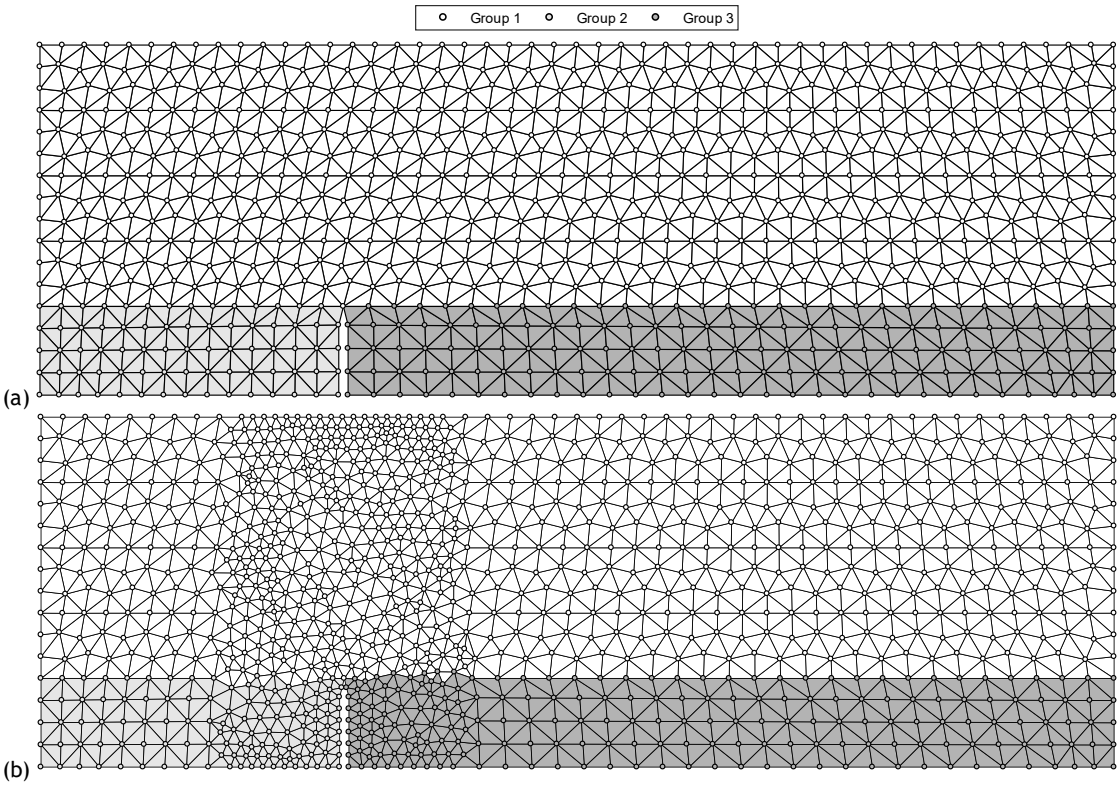

Figure 42 – Three point bending of a beam (a) 795 nodes mesh (b) 1076 nodes refined mesh

The results show that the predicted crack paths for every numerical method are very similar between each other and diverge slightly from the past paper result. However, the NNRPIM shows some an abrupt change to the right in the second to last iteration, which then continues in the following iterations. This change in direction that occurs in the NNRPIM is wrong, but excluding the results after that point, the crack path with the NNRPIM is close to the crack paths of the other numerical methods and Azevedo's work.

In the final step, using the FEM the number of nodes increased to 819 and the number of elements increased to 1480. Using the RPIM the number of nodes increased to 821 and the number of elements increased to 1484. Using the NNRPIM the number of nodes increased to 820 and the number of elements increased to 1482. With this mesh, this problem took 13 iterations to reach the final crack length.

A state of the mesh at the final iteration is presented in Figure 44, for each of the numerical methods used, except for the NNRPIM where the mesh presented is the mesh at the third to last iteration, the last iteration before the abrupt change in direction. Figure 44 shows that this algorithm, for this example and with this mesh deals well with the remeshing necessary for the crack propagation and almost all the elements have a good shape and are not distorted.

Regarding the groups assignments, they seem to be properly assigned for the most part, but the issue identified earlier about the elements with a different group from their nodes also occurs in this case.

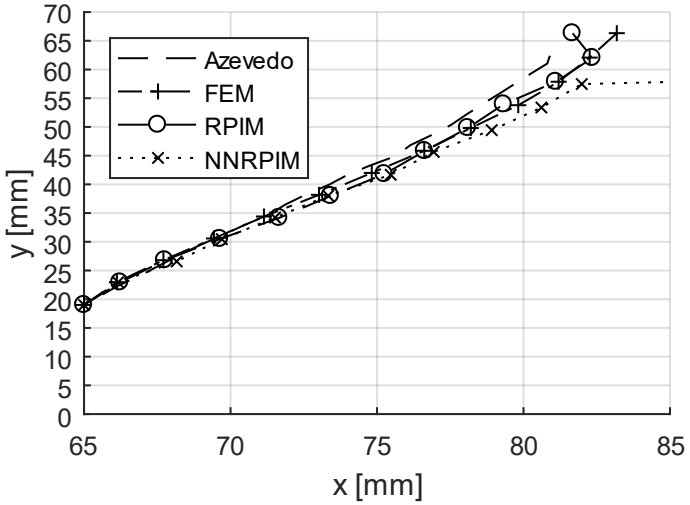

Figure 43 - Crack propagation for three point bending of a beam with 795 nodes mesh (axis not on the same scale)

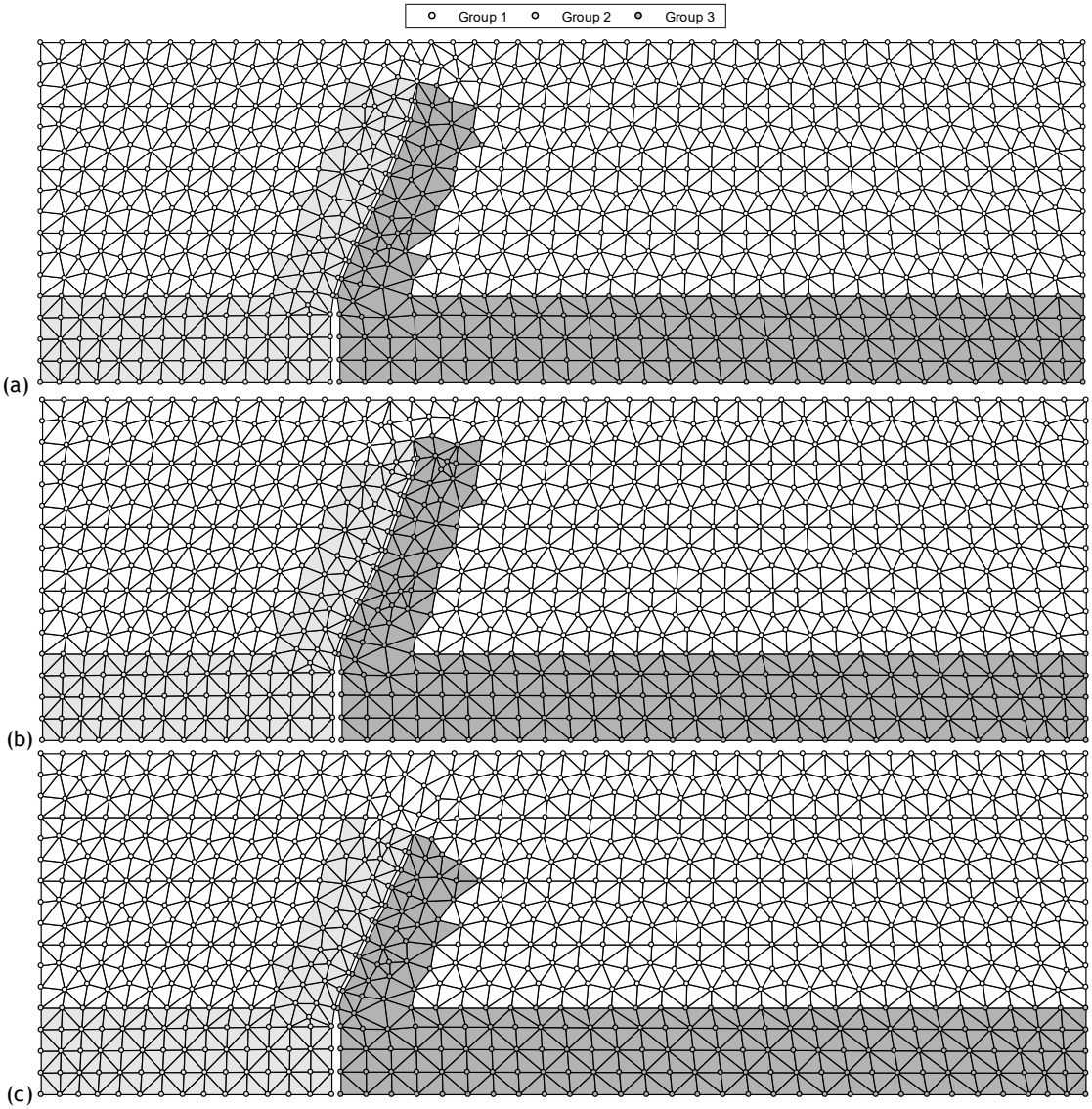

Figure 44 – Three point bending of a beam final mesh (a) FEM, (b) RPIM, (c) NNRPIM with 795 nodes initially

To verify the accuracy of results when refining the mesh in the area of the crack, this test was performed again for the refined mesh. This new mesh had initially 1076 nodes and 1992 elements, this represents an additional 281 nodes and 536 elements when compared to the unrefined mesh. The crack propagation was set to stop after the crack grew $50\ mm$, the same as for the unrefined mesh.

The results, seen in Figure 45, show that with this refinement the crack grows very similarly to the crack growth achieved by Azevedo. However, there is an abrupt change in direction at the last iteration when using NNRPIM, discarding that change, FEM and NNRPIM seem to be closest to the results achieved by Azevedo, while RPIM diverges a little, with a more pronounced curvature, but it is still very similar. As expected, the results prove to be better with this refined mesh, even if they were still very good with the original mesh in this example.

In the final step, the number of nodes increased to 1148 and the number of elements increased to 2090 using the FEM. Using the RPIM the number of nodes increased to 1146 and the number of elements increased to 2087. Using the NNRPIM the number of nodes increased to 1150 and the number of elements increased to 2094.

The total number of iterations to complete the propagation was 24, which is almost double the number of increments with the unrefined mesh, this was to be expected because Δa is approximately half of the Δa with the unrefined mesh. Notice that Δa is tied with the distance between the crack tip and the node of the crack closest to it.

A state of the mesh at the final iteration is shown in Figure 46, for each of the numerical methods used. Figure 46 shows that this algorithm, with the refined mesh, deals well with the remeshing necessary for the crack propagation and most elements have a good shape and are not distorted. Yet, in FEM there are some elements that are distorted, but the majority of elements are of good quality. This will affect the stress and strain calculation because FEM uses the elements to generate connectivity between nodes. But, as there are few distorted elements this effect is not enough to create incoherent results.

These distorted elements could possibly be avoided by also merging the nodes of the square in front of the crack with the nodes very close to them. But, during the development of the algorithm, this option led to many problems, with overlapping elements, and to the programming of complex special cases to ensure that element overlapping would not occur.

Figure 46c also demonstrates more clearly that in NNRPIM the crack did not grow as straight as when using the other numerical methods.

In terms of groups, once again, they were also attributed correctly with this more refined mesh, but there are also some cases where an element has all its nodes belonging to a group, while the element belongs to a different group.

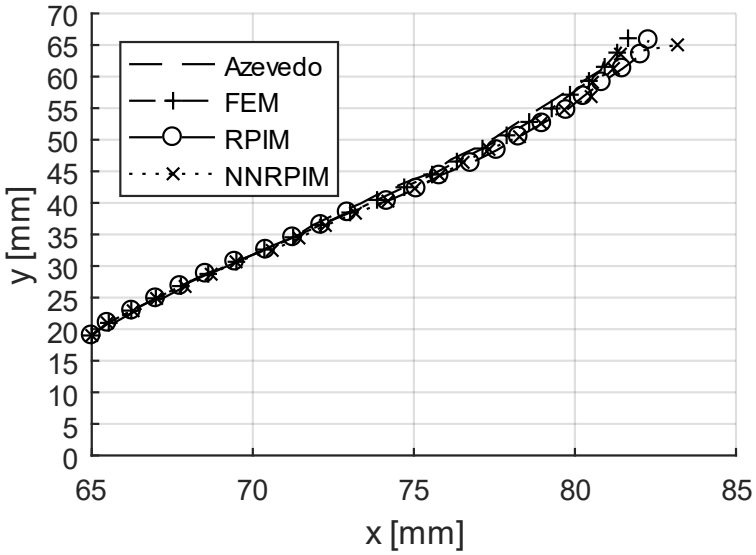

Figure 45 - Crack propagation for three point bending of a beam with 1076 nodes mesh (axis not on the same scale)

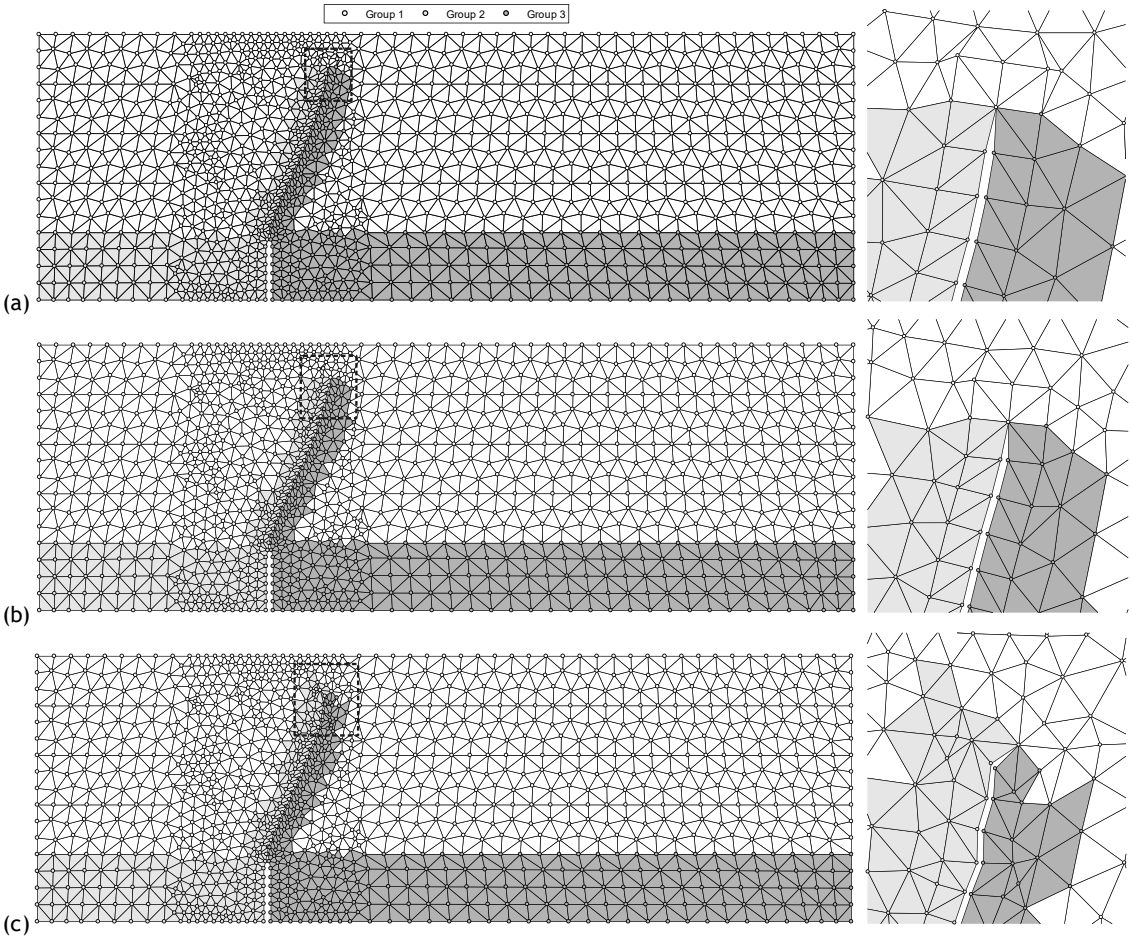

Figure 46 - Three point bending of a beam final mesh (a) FEM, (b) RPIM, (c) NNRPIM with 1076 nodes initially

6.5. Four Point Bending of a Beam with a Circular Hole

The fourth benchmark test conducted using this algorithm was a four point bending of a beam with a circular hole. The hole has a radius of $5.2\ mm$, the forces applied have a magnitude of $0.5\ N$ each and the material properties are $1000\ MPa$ and 0.29 for the Young's modulus and Poisson's ratio, respectively. The scheme with the measurements of this model is presented in Figure 47. This example has already been solved experimentally and numerically various times, including [4], [71], as a comparison with the results obtained here [4] was used.

As in previous examples, this problem was studied using a normal mesh and then a mesh refined in the crack area. The initial mesh, before the crack propagates further is seen in Figure 48a and the refined mesh is in Figure 48b.

The crack propagation path, when using the unrefined mesh is shown in Figure 49a, this mesh had initially 1112 nodes and 2047 elements and the crack was set to stop after propagating $14\ mm$.

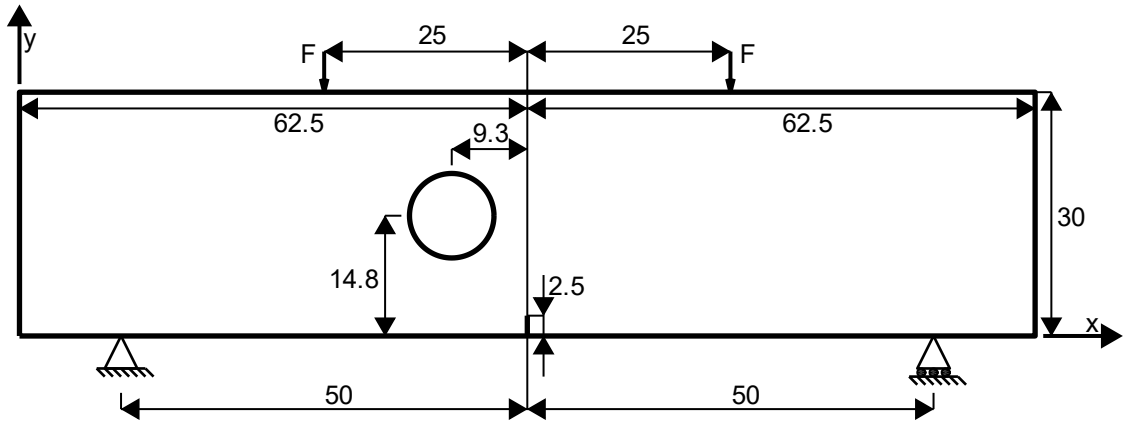

Figure 47 - Scheme of the four point bending of a beam with a circular hole example [measurements in mm]

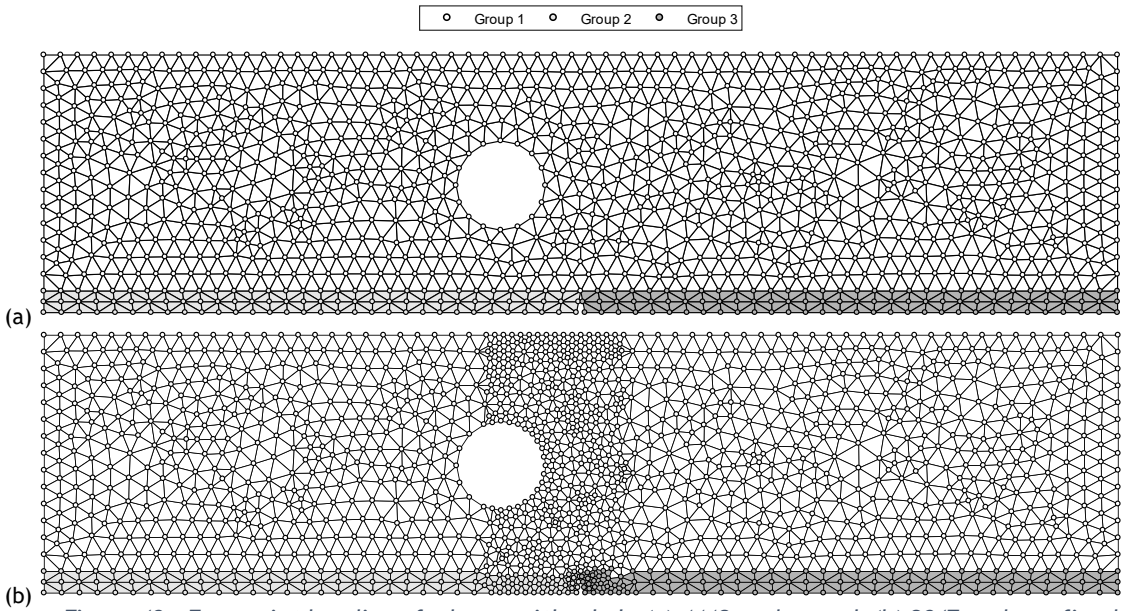

Figure 48 - Four point bending of a beam with a hole (a) 1112 nodes mesh (b) 2047 nodes refined mesh

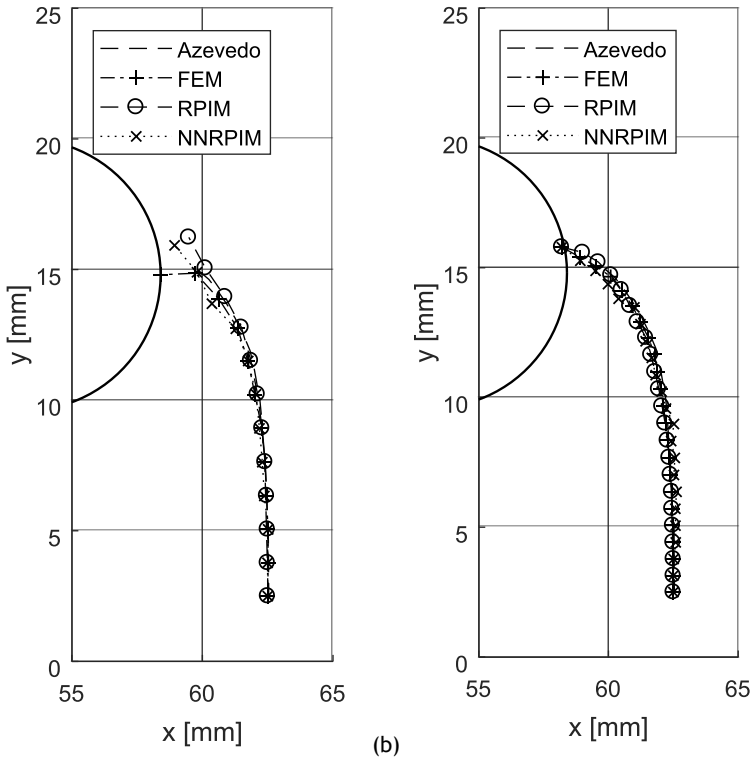

Figure 49 - Crack propagation for four point bending of a beam with a hole (a) with 1112 nodes mesh (b) 1354 nodes

The results show that the predicted crack paths for every numerical method are very similar between each other until approximately $y = 12.5\ mm$, after that, the NNRPIM behaves strangely, going under the results found in past papers by a considerable margin. The FEM results are good, except in the last iteration, where the crack goes abruptly to the left. Overall, using RPIM, the crack seems to grow very similarly to the growth found in past papers and it proves to have the best result using this unrefined mesh.

In the final step, using the FEM the number of nodes increased to 1145 and the number of elements increased to 2091. Using the RPIM, the number of nodes increased to 1143 and the number of elements increased to 2087. Using the NNRPIM the number of nodes also increased to 1143 and the number of elements increased to 2087. With this mesh, this problem took 12 iterations to reach the final crack length.

A state of the mesh at the final iteration is shown in Figure 50, for each of the numerical methods used, except for FEM, where the mesh shown is for the second to last iteration. Figure 50 shows that this algorithm, for this example and with this mesh, deals well with the remeshing necessary for the crack propagation. Most elements have a good shape and are not distorted, but there are a few elements in all the numerical methods that are a bit distorted. This is because this example is dealing with crack propagation in a region very near to a boundary, and in this algorithm the nodes of a boundary are prevented from moving in the process to achieve more uniform elements. Looking at the results, it is clear that this could be improved by moving the nodes of square surrounding the crack tip when the crack is close to a boundary, but this would require further development to ensure that the algorithm would behave properly with this exception.

Regarding the groups assignments, they seem to be properly assigned for the most part, but the issue identified earlier about the elements with a different group from their nodes also occurs in this case.

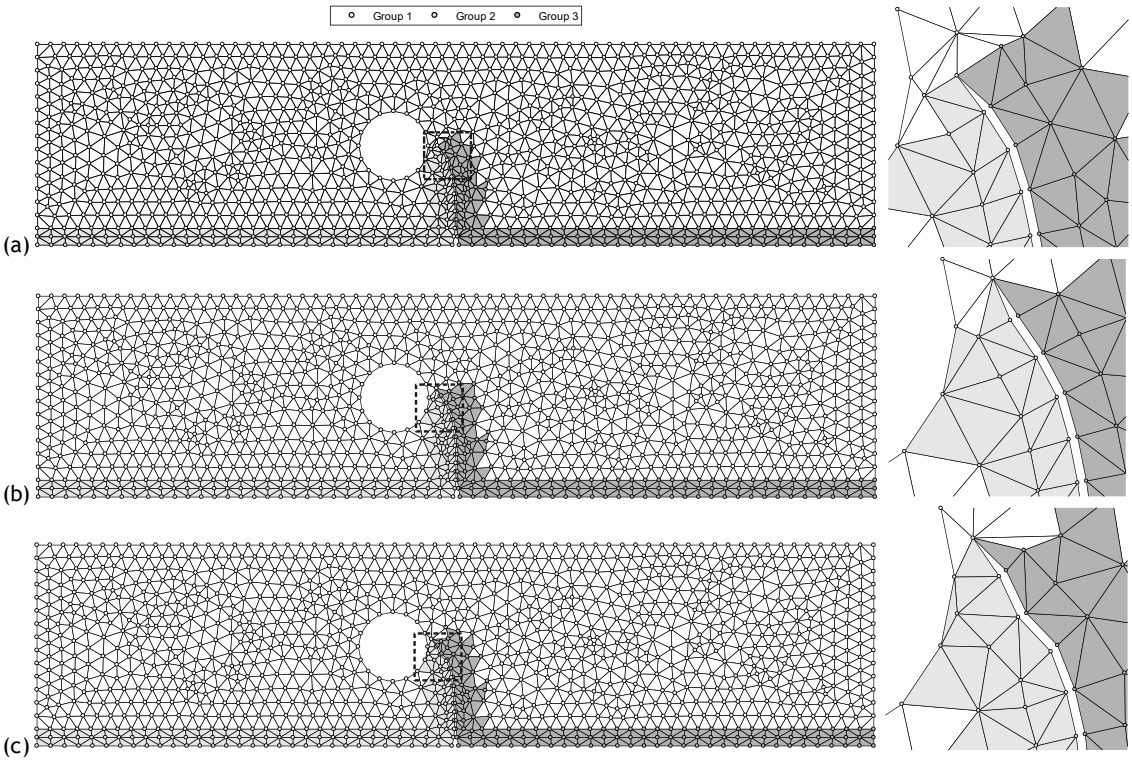

Figure 50 - Four point bending of a beam with a hole final mesh (a) FEM, (b) RPIM, (c) NNRPIM with 1112 nodes initially

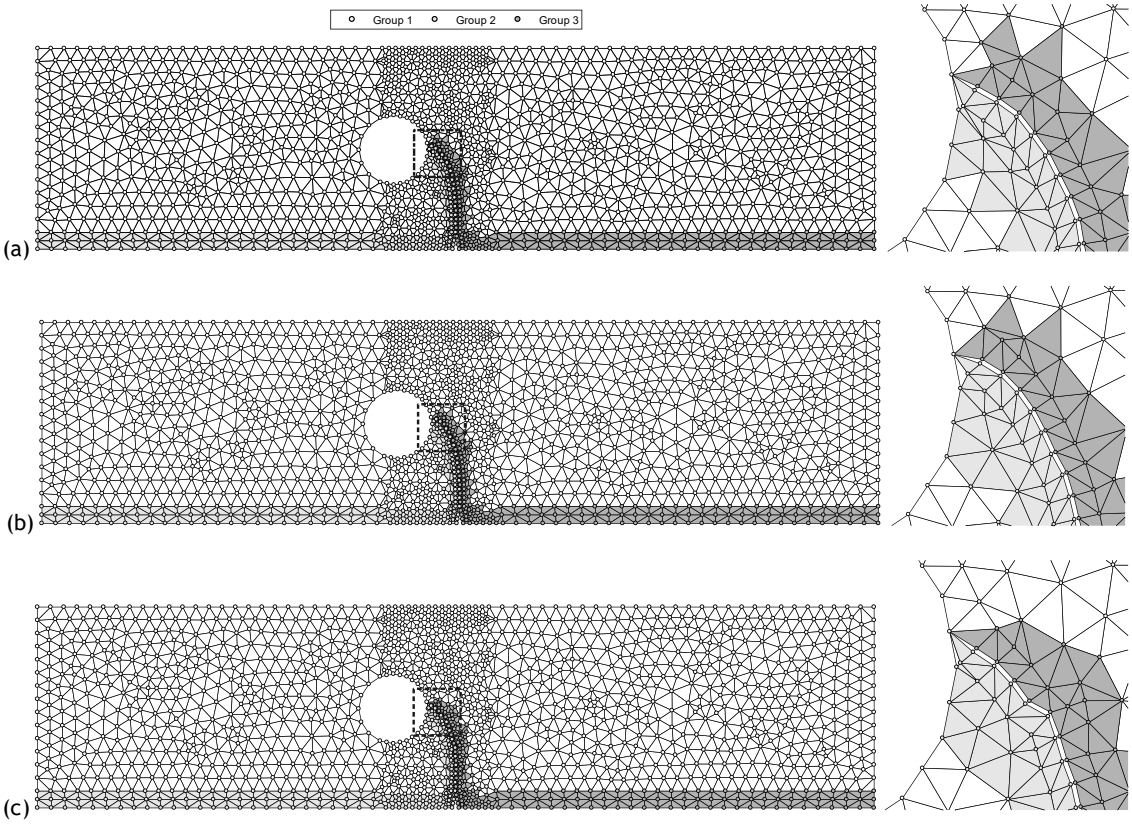

Figure 51 - Four point bending of a beam with a hole final mesh (a) FEM, (b) RPIM, (c) NNRPIM with 1354 nodes initially

Since the results were not very accurate with the previous mesh, a second test was done to verify the accuracy of results when refining the mesh in the area of the crack. This new

mesh had initially 1354 nodes and 2501 elements, this represents an additional 241 nodes and 454 elements when compared to the unrefined mesh. The crack propagation was set to stop after the crack grew $14.5\ mm$, $0.5\ mm$ more that in the unrefined mesh case, so it can reach a boundary node on the hole.

The results, presented in Figure 49b, show that with this refinement the crack grows very similarly to the crack growth achieved by Azevedo, independently of the numerical method used. These results are very good and prove that the crack propagation algorithm deals well with holes in the middle of the domain. There are some slight differences between the different numerical methods, but they are negligible and almost unnoticeable. The results are also much better than the results obtained with the unrefined mesh, as expected.

In the final step, the number of nodes increased to 1427 and the number of elements increased to 2603 when using the FEM. Using the RPIM the number of nodes increased to 1426 and the number of elements increased to 2601. Using the NNRPIM the number of nodes increased to 1427 and the number of elements increased to 2603.

The total number of iterations to complete the propagation was 23 which is almost double the number of increments with the unrefined mesh, this was to be expected because Δa is approximately half of the Δa with the unrefined mesh, because it is tied with the distance between the crack tip and the node of the crack closest to it.

A state of the mesh at the final iteration is presented in Figure 51, for each of the numerical methods used. Figure 51 shows that this algorithm, with the refined mesh, deals well with the presence of a hole inside the domain. As happened with the unrefined mesh, most elements have a good shape and are not distorted, but the same problem occurs as before, where some of the elements close to the hole are distorted due to the algorithm not being able to move the nodes of the boundary.

In terms of groups, once again, they were also attributed correctly with this more refined mesh but there are also some cases where an element has all its nodes belonging to a group, while the element belongs to a different group.

6.6. Mode II Inclined Crack

The fifth benchmark test performed with this algorithm was an inclined crack under mode II loading. The stresses applied have a magnitude of $5\ MPa$ each and the material properties are $1000\ MPa$ and 0.29 for the Young's modulus and Poisson's ratio, respectively. The scheme with the measurements of this model is presented in Figure 52. This example has been tested experimentally [72] and those experimental results are used as a comparison to the results obtained by the developed algorithm. It is important to do this test to verify how well the algorithm deals with an inclined crack and with a crack with two crack tips.

As in previous examples, this problem was studied using a normal mesh and then a mesh refined in the crack area. The initial mesh, before the crack propagates is seen in Figure 53a and the refined mesh is in Figure 53b. The crack was set to stop after propagating a combined $55\ mm$, i.e. $27.5\ mm$ on each side.

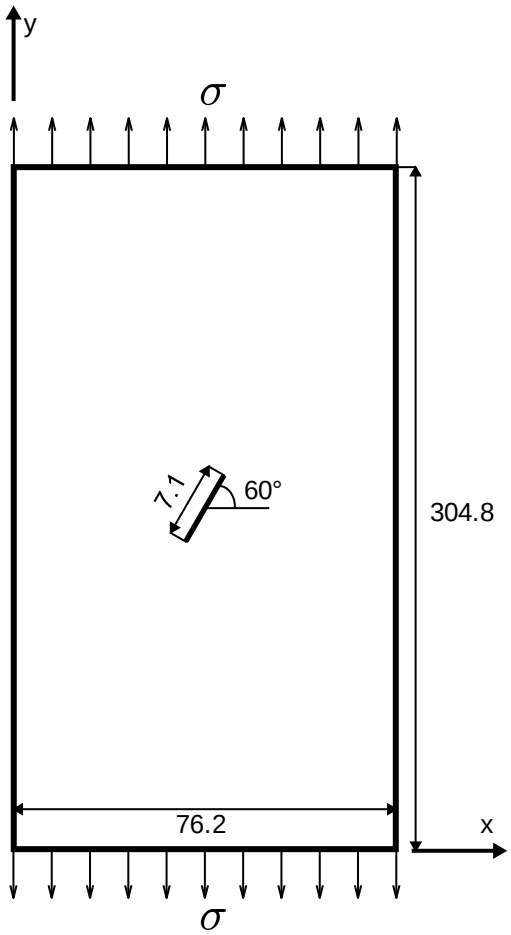

Figure 52 - Scheme of the mode II inclined crack example [measurements in mm] (not to scale)

The crack propagation path, when using the unrefined mesh is shown in Figure 54, this mesh had initially 550 nodes and 990 elements.

Due to the proximity of the two crack tips in the beginning of the propagation in this first mesh, the radius of the circle that determines which elements and nodes must change groups was altered, from triple the distance between the new crack tip and one of the edges of the square formed by the elements surrounding the new crack tip, to double that distance. This was done because with the triple, the group of one of the crack tips would be changed by the other crack tip in the first step since they are close to each other and the mesh is coarse, making the circle bigger.

The results show that the predicted crack paths for every numerical method are very similar, with small differences, but unlike the experimental results the crack paths achieved with this algorithm do not display an inflection to the centre of the plate, regardless of the numerical method. Another thing to note is that the crack paths of each crack tip are not exactly symmetric as should be expected. Overall the crack path obtained is different from the experimental crack path.

In the final step, for all the three methods the number of nodes increased to 556 and the number of elements decreased to 986. With this mesh, this problem took 5 iterations to reach the final crack length.

A state of the mesh at the final iteration is shown in Figure 55, for each of the numerical methods used. The minimal asymmetry of the crack paths observed before can be explained by the small differences in the remeshing of each crack tip as can be seen in Figure 55. Most elements have a good shape and are not distorted. Overall, Figure 55 shows that this algorithm

deals well with the remeshing necessary for a crack with two tips, but there can be some small differences in the remeshing of each crack tip, which should be the cause of the small asymmetries in the crack path.

Regarding the groups assignments, they seem to be properly assigned with the smaller radius of the circle of selection, but the issue identified earlier about the elements with a different group from their nodes also occurs in this case.

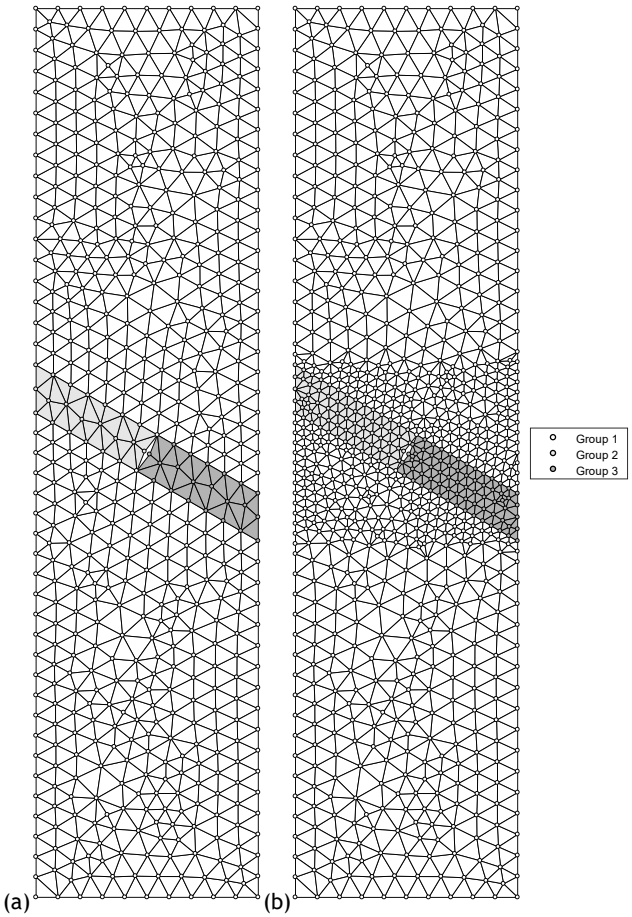

Figure 53 - Mode II inclined crack (a) 550 nodes mesh (b) 839 nodes refined mesh

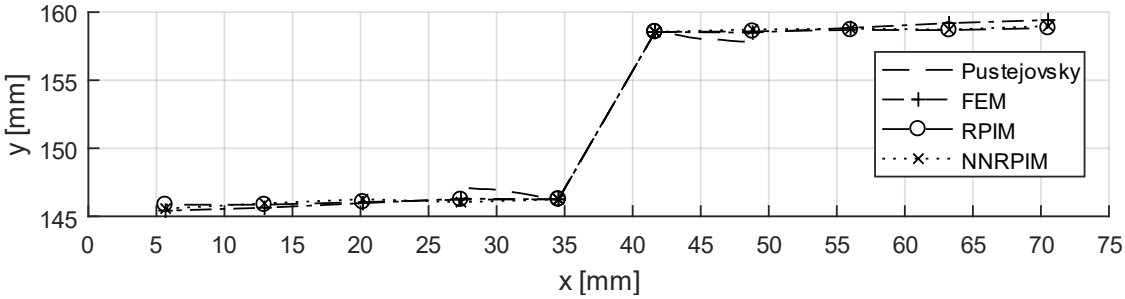

Figure 54 - Crack propagation for Mode II inclined crack with 550 nodes mesh

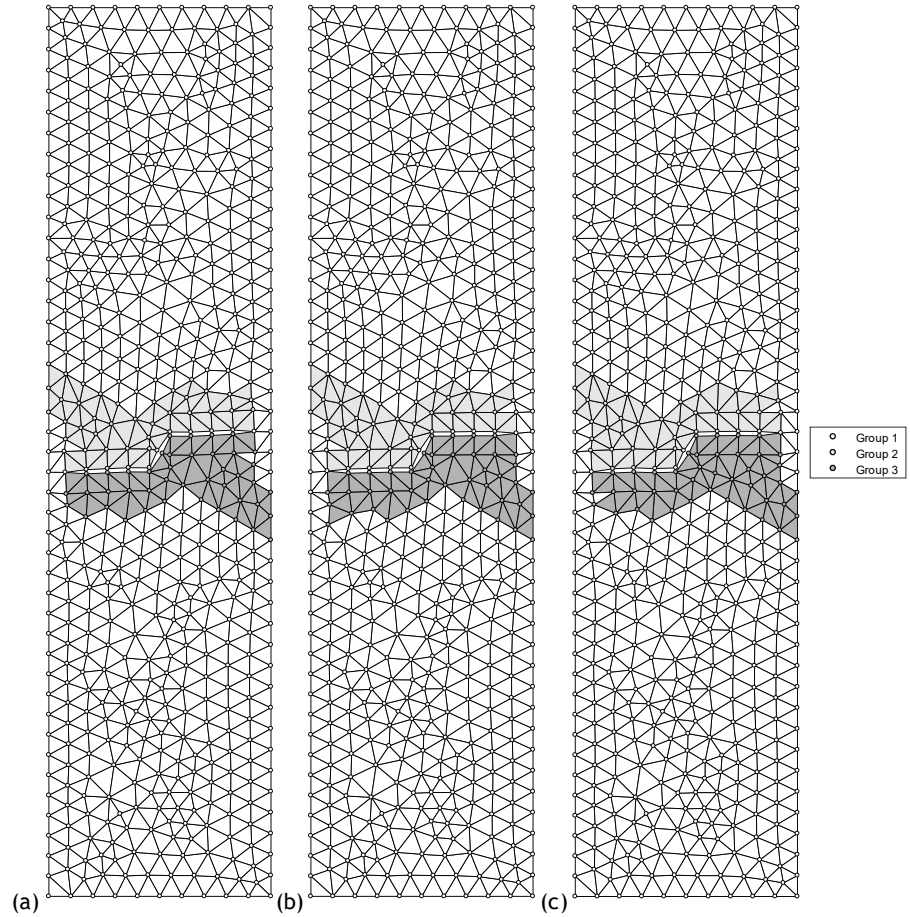

Figure 55 – Mode II inclined crack final mesh (a) FEM, (b) RPIM, (c) NNRPIM with 550 nodes initially

Since the results with the previous mesh were inaccurate, a second test with the mesh refined in the crack area was performed. This new mesh had initially 839 nodes and 1544 elements, this represents an additional 289 nodes and 554 elements when compared to the unrefined mesh. In this example, the radius of the circle to select the elements and nodes to change groups was changed back to the default triple the distance between the new crack tip and one of the edges of the square formed by the elements surrounding the new crack tip.

The results, presented in Figure 56, show that with this refinement the crack paths are slightly closer to the results achieved by Pustejovsky, when compared with the crack paths of the previous mesh, but they still remain very different from the experimental results and the refinement only provided a slight improvement of the results.

The assymetry of the crack paths of each crack tip has decreased and it is almost imperceptible with this refined mesh, this indicates that refined meshes seem to reduce the assimetry of the crack paths of the two crack tips.

In the final step, the number of nodes increased to 862 and the number of elements increased to 1558 when using the FEM and the RPIM. Using the NNRPIM the number of nodes increased to 560 and the number of elements increased to 1554.

The total number of iterations to complete the propagation was 9 which is slightly more than double the number of increments with the unrefined mesh, this was to be expected because Δa is approximately half of the Δa with the unrefined mesh, because it is tied with the distance between the crack tip and the node of the crack closest to it.

A state of the mesh at the final iteration is presented in Figure 57, for each of the numerical methods used. Figure 57 shows that this algorithm, with the refined mesh, deals well with

propagation of a crack with two tips. The FEM mesh and the RPIM mesh are very similar, however, it is interesting to verify that the NNRPIM mesh has two fewer elements because on the right crack tip there was a node deleted, which was not deleted in the other two numerical methods.

As happened with the unrefined mesh, most elements have a good shape and are not distorted, with some exceptions near the initial crack position.

In terms of groups, once again, they were also attributed correctly with this more refined mesh but there are also some cases where an element has all its nodes belonging to a group, while the element belongs to a different group.

In the end the crack paths for the tested numerical methods, independently of the mesh, behave as if the crack was subjected to mode I loading in this example, further tests have to be conducted to verify the exact causes of this and to improve the crack paths.

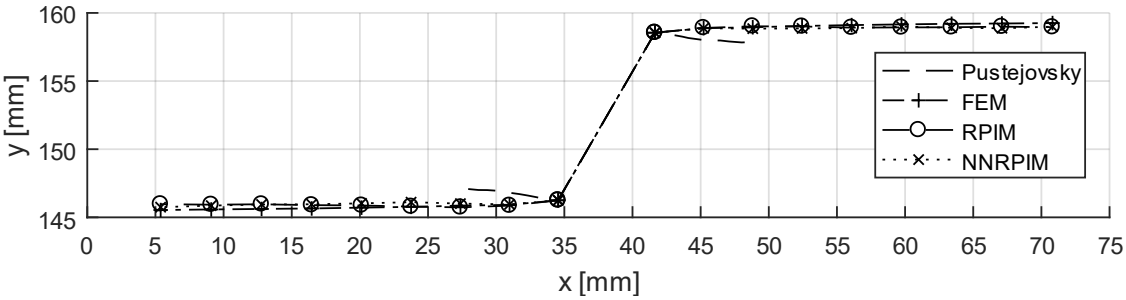

Figure 56 - Crack propagation for Mode II inclined crack with 839 nodes mesh

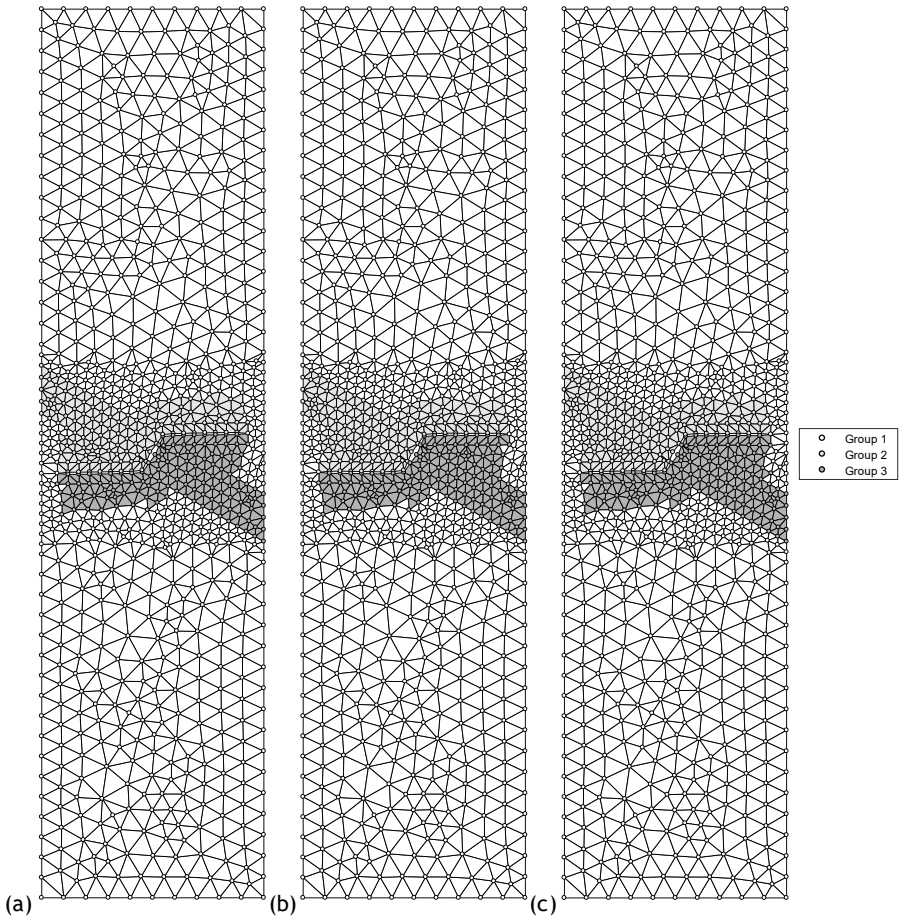

Figure 57 - Mode II inclined crack final mesh (a) FEM, (b) RPIM, (c) NNRPIM with 550 nodes initially

6.7. Plate with Two Edge Cracks and Two Holes

The sixth, and final, benchmark test performed with this algorithm was a plate with two edge cracks and two holes. A displacement of $0.1\ mm$ was imposed at the top and bottom boundaries of the plate and the material properties are $1000\ MPa$ and 0.3 for the Young's modulus and Poisson's ratio, respectively. The scheme with the measurements of this model is presented in Figure 58, the radius of the holes is $2\ mm$. This example has been solved numerically in the past [17], [73], with the work of Yang et al. being used as a comparison to the results obtained here. This is an important test to do because it will show how well the algorithm deals with separate cracks propagating in opposite directions.

Unlike previous examples, this problem was studied using a normal mesh and then a mesh refined in the whole domain, because in this case the crack propagates along most of the domain. The initial mesh, before the crack propagates, is seen in Figure 59a and the refined mesh is in Figure 59b. The crack was set to stop after propagating a combined $30\ mm$, i.e. $15\ mm$ on each side.

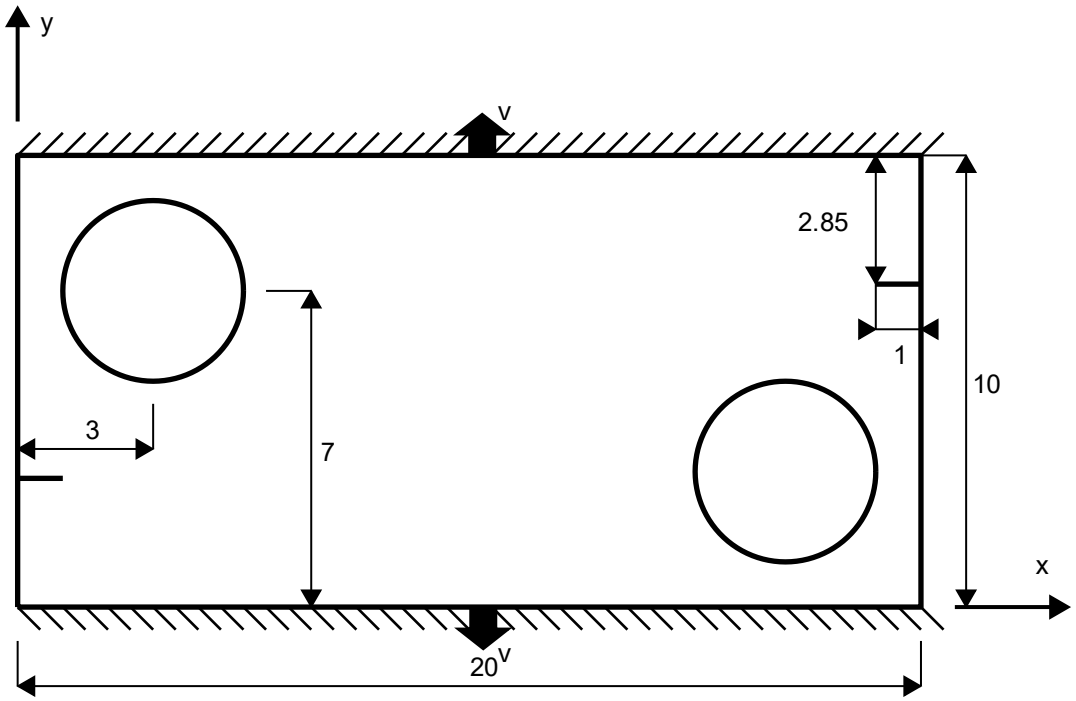

Figure 58 - Scheme of the plate with two edge cracks and two holes example [measurements in mm]

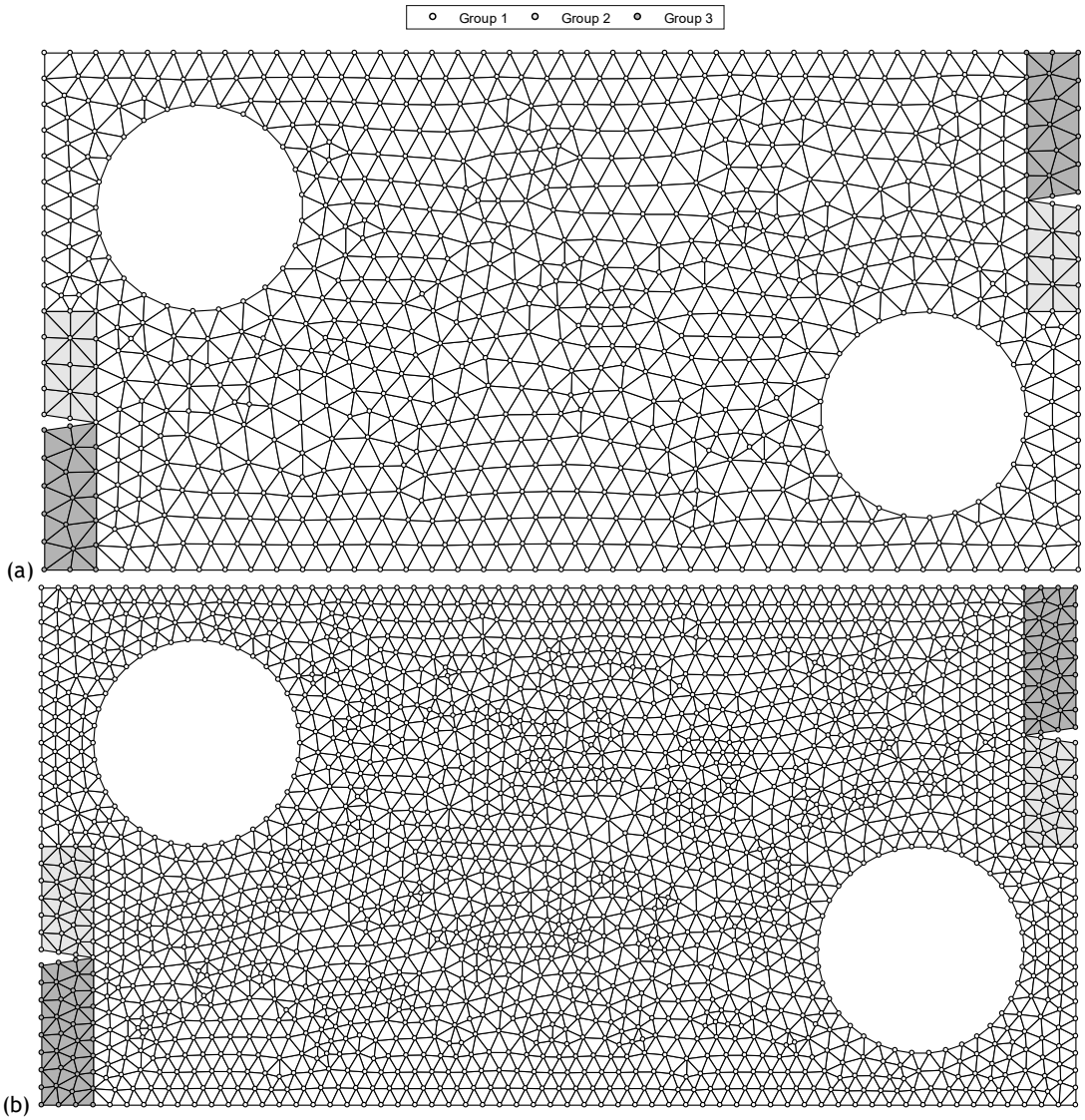

Figure 59 – Plate with two edge cracks and two holes (a) 820 nodes mesh (b) 1869 nodes refined mesh

The crack propagation paths, when using the unrefined mesh are shown in Figure 60, this mesh had initially 820 nodes and 1442 elements.

The results show that the crack paths differ slightly between the different numerical methods and, even though the crack paths' shape is similar to the ones achieved in previous works, they are not close to them. The only area where the crack paths have a different shape from the crack paths of the previous work is when the crack paths begin to approach the second hole, here, with this algorithm the cracks propagate towards the second hole, which did not happen to Yang et al.

The FEM crack paths are closest to the crack paths of previous works, while the NNRPIM crack paths are furthest from the crack paths of previous works. For all the numerical methods, the crack paths are mostly symmetric with some visible asymmetry near the second hole.

With the RPIM the cracks propagate until the last iteration without problems and with smooth curves, while with the FEM and the NNRPIM the cracks begin to propagate erratically when near the second hole.

In the final step, using the FEM the number of nodes increased to 854 and the number of elements decreased to 1442. Using the RPIM the number of nodes increased to 861 and the

number of elements decreased to 1436. Using the NNRPIM the number of nodes increased to 866 and the number of elements decreased to 1446. With this mesh, this problem took 29 iterations to reach the final crack length.

A state of the mesh at the last stable iteration is shown in Figure 61, for each of the numerical methods used. For the FEM and the NNRPIM this is the 25th iteration and for the RPIM this is the last iteration, as it propagated without problems.

Most elements have a good shape and are not distorted. Overall, Figure 61 shows that this algorithm deals well with the remeshing necessary for two cracks, but, as in the previous example, the propagation of two crack tips leads to small asymmetries in the mesh which affect the crack propagation slightly.

Regarding the groups assignments, they seem to be properly assigned considering the rules for that, but in the FEM and the NNRPIM meshes a limitation of this concept is observed. Due to the proximity of the cracks, there are some nodes that are in front of the crack tip but have the groups of one of the crack sides because the other crack had passed through that zone before. This means that some nodes are not interacting with nodes that they should be interacting with, which leads to inaccurate stress fields in that region. Changing the groups of those nodes to the crack tip group would mean that the nodes on the other side of the other crack could now see those nodes again, which is also incorrect. This could only be solved by using finer meshes or by adopting the visibility criterion.

This problem will not have an effect with the FEM, but it has an effect with the meshless methods, and it is possibly one of the causes of the erratic behaviour of the cracks near the second hole when using the NNRPIM. And it can also be the cause of the propagation in the direction of the second hole, which was not observed in past works.

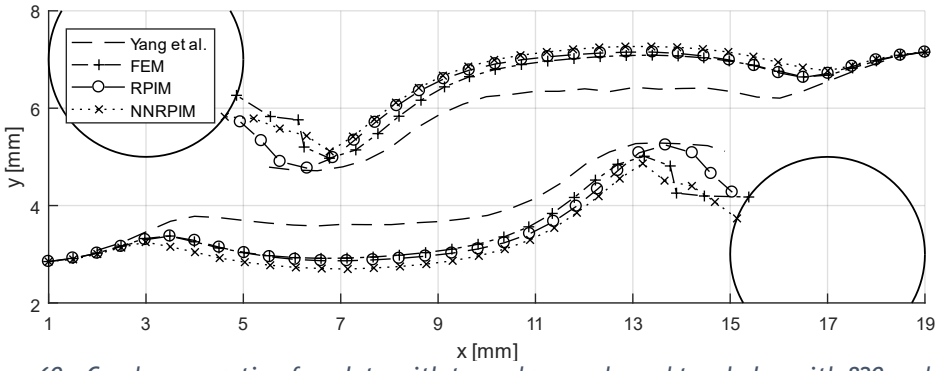

Figure 60 - Crack propagation for plate with two edge cracks and two holes with 820 nodes mesh

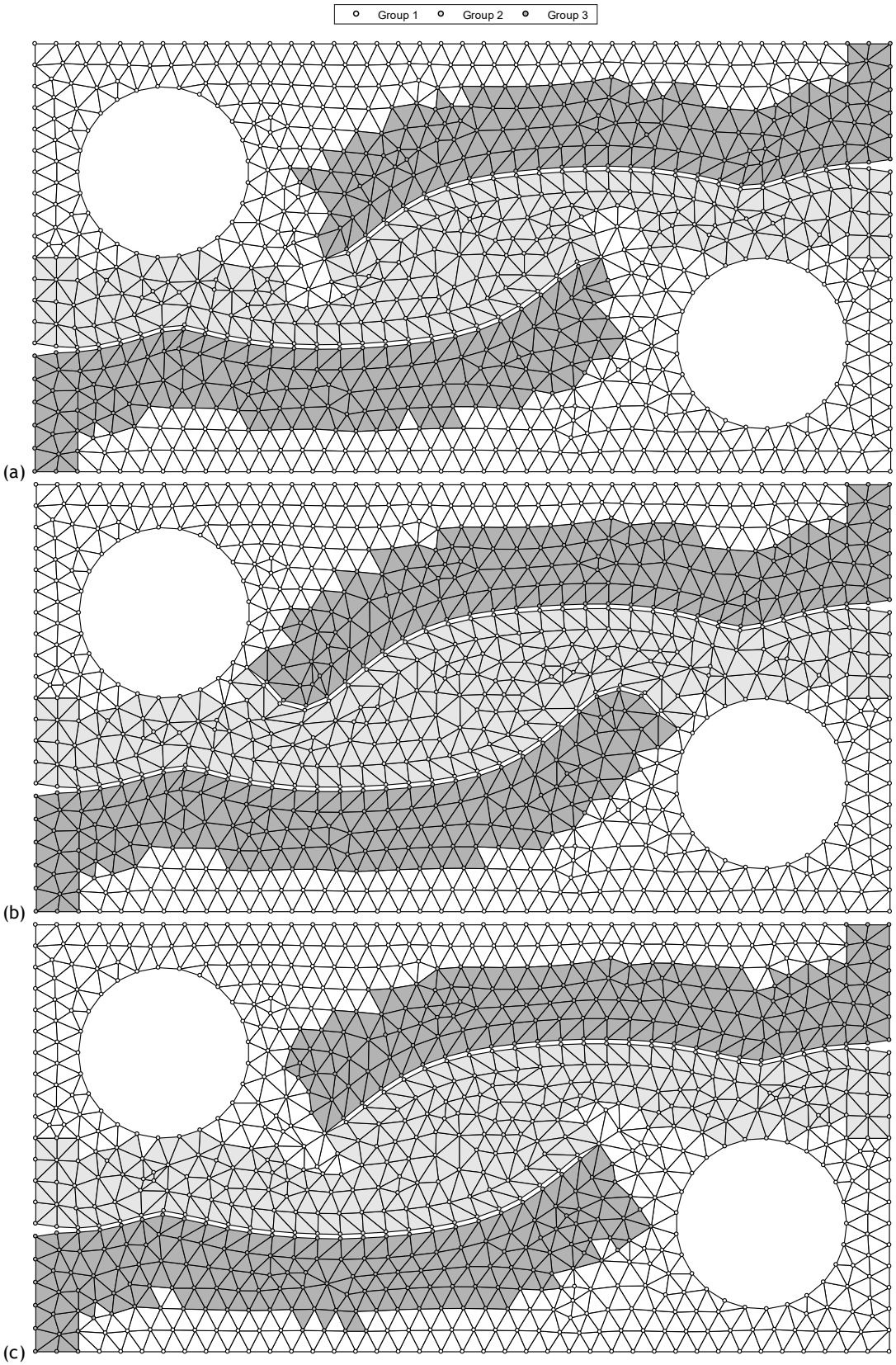

Figure 61 – Plate with two edge cracks and two holes final mesh (a) FEM, (b) RPIM, (c) NNRPIM with 820 nodes initially

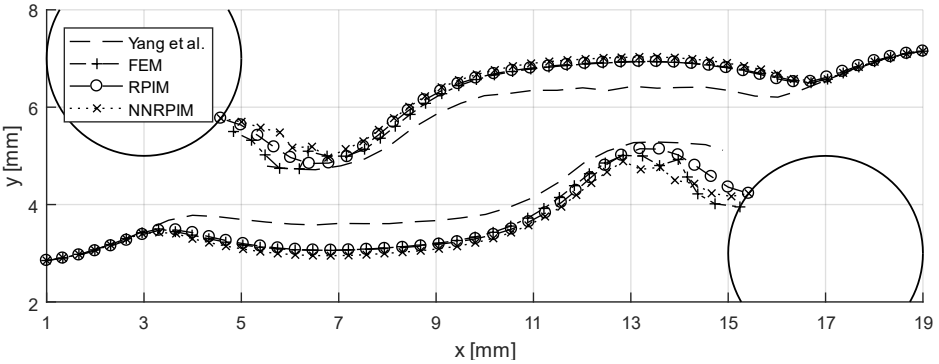

Figure 62 - Crack propagation for plate with two edge cracks and two holes with 1869 nodes mesh

Since the results with the previous mesh were inaccurate, a second test with the mesh refined in the whole domain was performed. This new mesh had initially 1869 nodes and 3472 elements, this represents an additional 1049 nodes and 2006 elements when compared to the unrefined mesh.

The results, presented in Figure 62, show that with this refinement the crack paths are slightly closer to the results achieved by Yang et al., when compared with the crack paths of the previous mesh, but they still remain far from them.

The erratic behaviour and the assymetry of the crack paths near the second hole when using the FEM and the NNRPIM also happens with this refined mesh. With this mesh the RPIM presents the best results, while the FEM and the NNRPIM are also very close but present those problems at the end of the crack propagation.

In the final step, using the FEM the number of nodes decreased to 1859 and the number of elements decreased to 3289. Using the RPIM the number of nodes decreased to 1858 and the number of elements decreased to 3286. Using the NNRPIM the number of nodes decreased to 1866 and the number of elements decreased to 3302. With this mesh, this problem took 42 iterations to reach the final crack length.

A state of the mesh at the last stable iteration is shown in Figure 63, for each of the numerical methods used. For the FEM this is the 37^{th} iteration, for the NNRPIM this is the 36^{th} iteration and for the RPIM this is the last iteration, as it propagated without problems.

Most elements have a good shape and are not distorted. Overall, Figure 63 shows that this algorithm deals well with the remeshing necessary for two cracks, but, as occurred with the previous mesh, the propagation of two cracks tips leads to small asymmetries in the mesh which affect the crack propagation slightly.

Regarding the groups assignments, they seem to be properly assigned considering the rules for that, but even with this refined mesh the same problem of the visibilities identified for the previous mesh is verified here, this means that an even finer mesh is needed to solve this.

This problem is still possibly one of the causes of the erratic behaviour of the cracks near the second hole when using the NNRPIM, and also one of the causes of the propagation towards the second hole that happens with the RPIM.

These results indicate that with further mesh refinement the crack paths should converge with the crack paths found in previous works, unfortunately it was not possible to perform tests with more refined meshes due to time constrains.

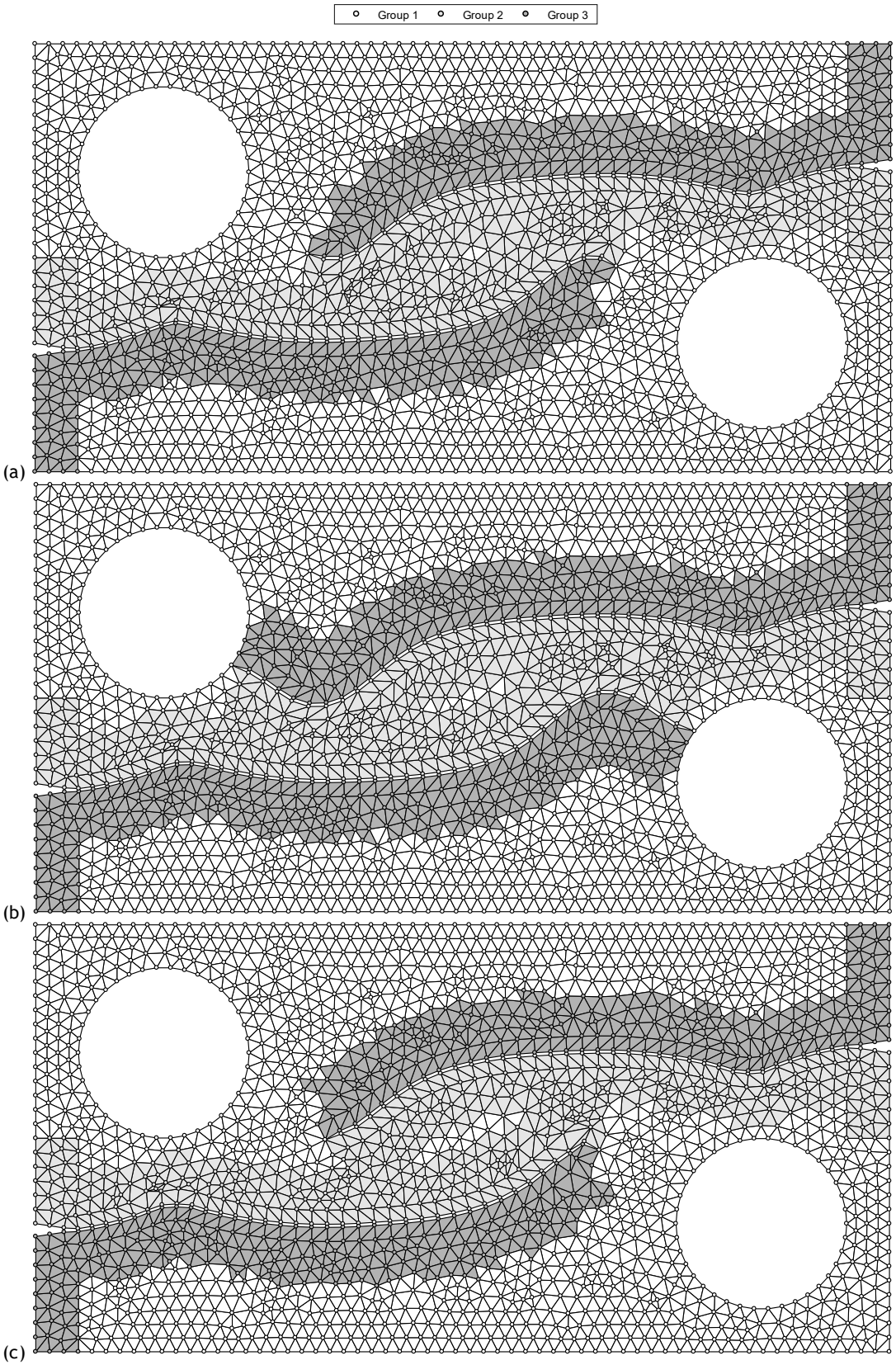

Figure 63 - Plate with two edge cracks and two holes final mesh (a) FEM, (b) RPIM, (c) NNRPIM with 1869 nodes initially

Chapter 7

Conclusion

In this work, an algorithm for crack propagation was developed, this algorithm is completely automatic, the user only has to define the domain, the boundary conditions and the maximum crack length. Then, the algorithm calculates the crack growth iteratively until the maximum crack length is reached or a boundary is reached. This algorithm works with three different numerical methods, the finite element method, the radial point interpolation method and the natural neighbours radial point interpolation method.

One requisite of this algorithm is that the crack increment Δa is tied with the size of the mesh. Another requisite was the related with the universality of the algorithm: it must work regardless of the numerical method used. Due to these two requisites, elements had to be used in this algorithm to accommodate for the FEM and elements are difficult to work with when dealing with crack propagation, they either must be cut, leading to sometimes very distorted elements, or remeshed, or some other option shown in section 3.1. So, to have fewer distorted elements, it was decided to tie the crack growth with size of the mesh by setting it was equal to the distance between the crack tip node and the node of the crack closest to the crack tip, excluding the crack tip itself.

An advantage of the algorithm presented here is that the number of nodes and elements does not increase by much as the crack grows, unlike some other solutions developed for crack propagation, this is achieved while also obtaining accurate results. And while the total number of elements and nodes does not increase by much, this algorithm also ensures that there are always eight elements around the crack.

The disposition of the elements and nodes around the crack tip ensures that there are always eight elements around it. This leads to a more accurate crack tip stress calculation in the step where the propagation direction is determined. If there were fewer elements around the crack tip, there would be fewer integration point from which to obtain the stress figures to calculate that weighted average.

Regarding the different numerical methods used, with this algorithm it was shown that the solutions for the crack path prediction were generally very close to the solutions found in literature for all the methods. The only exceptions to this were the last two examples, but it seems that would converge with the results found in literature using finer meshes, the mode II inclined crack example does not seem to begin converging using finer meshes. Excluding the

mode I and the three point bending of a beam, the solutions using very coarse meshes diverged by a visible margin from the solutions of past research works.

In the four point bending of a beam with a hole, this was mostly due to the hole which caused some remeshing problems and there is also a visible curvature in the crack path, due to this curvature it is advisable to have a smaller Δa. In this algorithm, this is achieved by constructing a more refined mesh around the crack tip at the beginning of the analysis. Therefore, it is advisable to have refined meshes in the crack area when dealing with cracks where their path is expected to be curved or pass near a hole.

In the mode II shear loading example, it was because this was the coarsest mesh used in this work for the mixed mode case, so it showed that there is a limit for how coarse a mesh can be and still provide accurate crack paths when using this algorithm. Therefore, the author would advise having at least around 700 nodes when dealing with mixed mode loading problems.

The inaccurate results of the mode II inclined crack example could possibly be solved by using a different propagation criterion, not dependent on the stress fields.

In the end, the algorithm proved to be very robust and it worked without numerical problems, even with coarse meshes. It also showed that with appropriate meshes this algorithm can solve a variety of problems accurately and that it can solve problems with multiple crack tips.

7.1. Further improvements

As already mentioned in the results discussion, the algorithm could be improved by allowing the nodes surrounding the crack tip to be moved when one of the connector nodes is in a boundary, this could improve the mesh quality when dealing with problems with holes inside the domain by having fewer distorted elements.

Another interesting expansion to this algorithm would be using different propagation criteria to compare with the results obtained while using the maximum tangential stress criterion. It would be especially interesting to use this algorithm with an energy-based criterion and a strain-based criterion, so all three types of criteria could be compared.

Since triangular elements with a single integration point at their centre are not the best type of elements in the FEM, for the crack tip elements these elements can be converted into quarter-tip elements. This will involve deleting the extra nodes once the crack propagated to a new position, so it would require some more developing time to ensure that this will not cause any problems.

Additionally, it would be interesting to consider the visibility in meshless using either the visibility criterion, the diffraction criterion or the transparency criterion, because the boundary nodes are known, and this includes the crack nodes, removing the concept of groups. This makes it possible to know if a straight line from an integration point to a node would go through a boundary, excluding that node from the influence domain of that integration point, in the visibility criterion, or changing its weight in the other two. This change could improve slightly the results obtained using the meshless methods. It was not performed in the scope of this thesis due to time constrains.

A final improvement that could also be made to this algorithm is to have the crack stop growing automatically, when the stresses at the tip are lower than the failure stress, instead of it stopping when reaching a specified length.

Expansion to material non-linearity, plasticity, this could pose some problems because nodes and elements are deleted and remeshed at each step, a process to extrapolate the residual stresses and strains for the new nodes would have to be created, otherwise the algorithm could adapt well to material non-linearity.

Expanding this algorithm to a 3D space would be hard, due to how the connector elements are created, it would mean that the rules for their creation would have to be completely reworked. Moving past that difficulty the rest of the algorithm would be easy to adapt.

References

[1] J. Belinha, *Meshless Methods in Biomechanics*. Springer, 2014.

[2] J.-S. Chen, M. Hillman, and S.-W. Chi, "Meshfree Methods: Progress Made after 20 Years," *J. Eng. Mech.*, vol. 143, no. 4, p. 04017001, 2017.

[3] V. P. Nguyen, T. Rabczuk, S. Bordas, and M. Duflot, "Meshless methods: A review and computer implementation aspects," *Math. Comput. Simul.*, vol. 79, no. 3, pp. 763–813, 2008.

[4] J. M. C. Azevedo, "Fracture mechanics using the Natural Neighbour Radial Point Interpolation Method," Faculdade de Engenharia da Universidade do Porto, 2013.

[5] J. G. Wang and G. R. Liu, "A point interpolation meshless method based on radial basis functions," *Int. J. Numer. Methods Eng.*, vol. 54, no. 11, pp. 1623–1648, 2002.

[6] M. Kuna, *Finite Elements in Fracture Mechanics*. Springer, 2010.

[7] "Why bother if they are the same?," *University of Cambridge*. [Online]. Available: https://www.doitpoms.ac.uk/tlplib/brittle_fracture/same.php. [Accessed: 10-Jan-2018].

[8] C. E. Inglis, "Stresses in a plate due to the presence of cracks and sharp corners," *Trans. Inst. Naval Arch.*, vol. 55. pp. 219–239, 1913.

[9] A. A. Griffith; and M. Eng, "VI. The phenomena of rupture and flow in solids," *Philos. Trans. R. Soc. London. Ser. A, Contain. Pap. a Math. or Phys. Character*, vol. 221, no. 582–593, p. 163 LP-198, Jan. 1920.

[10] H. M. Westergaard, "Bearing pressures and cracks," *J. Appl. Mech.*, vol. 6, pp. A49–A53, 1939.

[11] G. R. Irwin, "Analysis of Stresses and Strains Near the End of a Crack Traversing a Plate," *Journal of Applied Mechanics*, vol. 24, no. September. pp. 361–364, 1957.

[12] D. Salimi-Majd, F. Shahabi, and B. Mohammadi, "Effective local stress intensity factor criterion for prediction of crack growth trajectory under mixed mode fracture conditions," *Theor. Appl. Fract. Mech.*, vol. 85, pp. 207–216, 2016.

[13] M. M. Mirsayar, "Mixed mode fracture analysis using extended maximum tangential strain criterion," *Mater. Des.*, vol. 86, pp. 941–947, 2015.

[14] M. A. Hussain, S. L. Pu, and J. Underwood, "Strain energy release rate for a crack under combined mode I and mode II," in *Fracture Analysis: Proceedings of the 1973 National Symposium on Fracture Mechanics, Part II*, 1974.

[15] G. C. Sih, "Strain-energy-density factor applied to mixed mode crack problems," *Int. J. Fract.*, vol. 10, no. 3, pp. 305–321, 1974.

[16] F. Erdogan and G. C. Sih, "On the Crack Extension in Plates Under Plane Loading and Transverse Shear," *J. Basic Eng.*, vol. 85, no. 4, p. 519, 1963.

[17] P. O. Bouchard, F. Bay, and Y. Chastel, "Numerical modelling of crack propagation: Automatic remeshing and comparison of different criteria," *Comput. Methods Appl. Mech. Eng.*, vol. 192, no. 35–36, pp. 3887–3908, 2003.

[18] Kaung Jain Chang, "On the maximum strain criterion-a new approach to the angled crack problem," *Eng. Fract. Mech.*, vol. 14, no. 1, pp. 107–124, 1981.

[19] J. R. Rice, "A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notches and Cracks," *J. Appl. Mech.*, vol. 35, no. 2, p. 379, 1968.

[20] G. P. Cherepanov, "Crack propagation in continuous media," *J. Appl. Math. Mech.*, vol. 31, no. 3, pp. 503–512, 1967.

[21] L. Banks-Sills, "Application of the Finite Element Method to Linear Elastic Fracture Mechanics," *Appl. Mech. Rev.*, vol. 44, no. 10, pp. 447–461, 1991.

[22] F. Z. Li, C. F. Shih, and A. Needleman, "A comparison of methods for calculating energy release rates," *Eng. Fract. Mech.*, vol. 21, no. 2, pp. 405–421, 1985.

[23] M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp, "Stiffness and Deflection Analysis of Complex Structures," *J. Aeronaut. Sci.*, vol. 23, no. 9, pp. 805–823, 854, 1956.

[24] R. W. Clough and E. L. Wilson, “Early Finite Element Research at Berkeley,” *The Fifth U.S. National Conference on Computational Mechanics*. 2000.
[25] R. Clough, “The stress distribution of Norfork Dam,” 1962.
[26] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method Volume 1 : The Basis*, 5th ed. Wiley John & Sons, 2000.
[27] Y. Duan, “A note on the meshless method using radial basis functions,” *Comput. Math. with Appl.*, vol. 55, no. 1, pp. 66–75, 2008.
[28] L. B. Lucy, “A numerical approach to the testing of the fission hypothesis,” *Astron. J.*, vol. 82, no. December, pp. 1013–1024, 1977.
[29] R. A. Gingold and J. J. Monaghan, “Smoothed particle hydrodynamics: theory and application to non-spherical stars,” *Mon. Not. R. Astron. Soc.*, vol. 181, no. 3, pp. 375–389, 1977.
[30] W. Benz and E. Asphaug, “Simulations of brittle solids using smooth particle hydrodynamics,” *Comput. Phys. Commun.*, vol. 87, no. 1–2, pp. 253–265, 1995.
[31] T. Belytschko, Y. Guo, W. K. Liu, and S. P. Xiao, “A unified stability analysis of meshless particle methods,” *Int. J. Numer. Methods Eng.*, vol. 48, no. 9, pp. 1359–1400, 2000.
[32] J. W. Swegle, D. L. Hicks, and S. W. Attaway, “Smoothed Particle Hydrodynamics Stability Analysis,” *J. Comput. Phys.*, vol. 116, no. 1, pp. 123–134, 1995.
[33] P. Lancaster and K. Salkauskas, “Surface generated by moving least square methods,” *Math. Comput.*, vol. 37, no. 155, pp. 141–158, 1981.
[34] B. Nayroles, G. Touzot, and P. Villon, “Generalizing the finite element method: Diffuse approximation and diffuse elements,” *Comput. Mech.*, vol. 10, no. 5, pp. 307–318, 1992.
[35] T. Belytschko, Y. Y. Lu, and L. Gu, “Element-free Galerkin methods,” *Int. J. Numer. Methods Eng.*, vol. 37, no. 2, pp. 229–256, 1994.
[36] T. Belytschko, Y. Y. Lu, and L. Gu, “Crack propagation by element-free Galerkin methods,” *Eng. Fract. Mech.*, vol. 51, no. 2, pp. 295–315, 1995.
[37] T. Belytschko, D. Organ, and Y. Krongauz, “A coupled finite element-element-free Galerkin method,” *Comput. Mech.*, vol. 17, no. 3, pp. 186–195, 1995.
[38] W. K. Liu, S. Jun, and Y. F. Zhang, “Reproducing kernel particle methods,” *Int. J. Numer. Methods Fluids*, vol. 20, no. 8–9, pp. 1081–1106, 1995.
[39] S. N. Atluri and T. Zhu, “A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics,” *Comput. Mech.*, vol. 22, no. 2, pp. 117–127, 1998.
[40] J. Nitsche, “Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind,” *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*. Springer, Berlin, pp. 9–15, 1971.
[41] M. Griebel and M. A. Schweitzer, “A Particle-Partition of Unity Method-Part IV: Parallelization,” in *Meshfree Methods for Partial Differential Equations*, M. Griebel and M. A. Schweitzer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 161–192.
[42] I. Kaljević and S. Saigal, “An improved element free Galerkin formulation,” *Int. J. Numer. Methods Eng.*, vol. 40, no. 16, pp. 2953–2974, 1997.
[43] J. Braun and M. Sambridge, “A numerical method for solving partial differential equations on highly irregular evolving grids,” *Nature*, vol. 376, no. 6542, pp. 655–660, 1995.
[44] N. Sukumar, B. Moran, and T. Belytschko, “The natural element method in solid mechanics,” *Int. J. Numer. Methods Eng.*, vol. 43, no. 5, pp. 839–887, 1998.
[45] G. Voronoi, “Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs.,” *J. für die reine und Angew. Math.*, vol. 134, pp. 198–287, 1908.
[46] B. Delaunay, “Sur la sphère vide. A la mémoire de Georges Voronoï,” *Bull. l’Académie des Sci. l’URSS*, no. 6, pp. 793–800, 1934.
[47] G. R. Liu and Y. T. Gu, “A point interpolation method for two-dimensional solids,” *Int. J. Numer. Methods Eng.*, vol. 50, no. 4, pp. 937–951, 2001.
[48] G. R. Liu, “A point assembly method for stress analysis for two-dimensional solids,” *Int. J. Solids Struct.*, vol. 39, no. 1, pp. 261–276, 2002.
[49] L. M. J. S. Dinis, R. M. Natal Jorge, and J. Belinha, “Analysis of 3D solids using the natural neighbour radial point interpolation method,” *Comput. Methods Appl. Mech.*

Eng., vol. 196, no. 13–16, pp. 2009–2028, 2007.
[50] J. Belinha, L. M. J. S. Dinis, and R. M. N. Jorge, "The natural neighbour radial point interpolation method: dynamic applications," Faculdade de Engenharia da Universidade do Porto, 2010.
[51] J. M. C. Azevedo, J. Belinha, L. M. J. S. Dinis, and R. M. Natal Jorge, "Crack path prediction using the natural neighbour radial point interpolation method," *Eng. Anal. Bound. Elem.*, vol. 59, pp. 144–158, 2015.
[52] J. Belinha, J. M. C. Azevedo, L. M. J. S. Dinis, and R. M. Natal Jorge, "The Natural Neighbor Radial Point Interpolation Method Extended to the Crack Growth Simulation," *Int. J. Appl. Mech.*, vol. 08, no. 01, p. 1650006, 2016.
[53] J. G. Wang and G. R. Liu, "On the optimal shape parameters of radial basis functions used for 2-D meshless methods," *Comput. Methods Appl. Mech. Eng.*, vol. 191, no. 23–24, pp. 2611–2630, 2002.
[54] P. O. Bouchard, F. Bay, Y. Chastel, and I. Tovena, "Crack propagation modelling using an advanced remeshing technique," *Comput. Methods Appl. Mech. Eng.*, vol. 189, no. 3, pp. 723–742, 2000.
[55] G. Dhondt, "Cutting of a 3-D finite element mesh for automatic mode I crack propagation calculations," *Int. J. Numer. Methods Eng.*, vol. 42, no. 4, pp. 749–772, 1998.
[56] D. Colombo and M. Giglio, "A methodology for automatic crack propagation modelling in planar and shell FE models," *Eng. Fract. Mech.*, vol. 73, no. 4, pp. 490–504, 2006.
[57] R. D. Henshell and K. G. Shaw, "Crack tip finite elements are unnecessary," *Int. J. Numer. Methods Eng.*, vol. 9, no. 3, pp. 495–507, 1975.
[58] S. Soghrati, F. Xiao, and A. Nagarajan, "A conforming to interface structured adaptive mesh refinement technique for modeling fracture problems," *Comput. Mech.*, vol. 59, no. 4, pp. 667–684, 2017.
[59] K. Park, G. H. Paulino, W. Celes, and R. Espinha, "Adaptive mesh refinement and coarsening for cohesive zone modeling of dynamic fracture," *Int. J. Numer. Methods Eng.*, vol. 92, no. 1, pp. 1–35, 2012.
[60] N. Moës, J. Dolbow, and T. Belytschko, "A finite element method for crack growth without remeshing," *Int. J. Numer. Meth. Engng*, vol. 46, no. 1, pp. 131–150, 1999.
[61] P. M. A. Areias and T. Belytschko, "Analysis of three-dimensional crack initiation and propagation using the extended finite element method," *Int. J. Numer. Methods Eng.*, vol. 63, no. 5, pp. 760–788, 2005.
[62] J. G. G. C. Ferreira, "Numerical Modelling of Fatigue Crack Growth Using XFEM," Universidade de Aveiro, 2014.
[63] E. B. Chin, J. B. Lasserre, and N. Sukumar, "Modeling crack discontinuities without element-partitioning in the extended finite element method," *Int. J. Numer. Methods Eng.*, vol. 110, no. 11, pp. 1021–1048, 2017.
[64] T. Belytschko, L. Gu, Y. Y. Lu, L. G. T. Belytschko, and Y. Y. Lu, "Fracture and crack growth by element-free Galerkin methods.," *Model. Simul. Mater. Sci. Engrg.*, vol. 2, pp. 519–534, 1994.
[65] T. Belytschko, Y. Y. Lu, L. Gu, and M. Tabbara, "Element-free galerkin methods for static and dynamic fracture," *Int. J. Solids Struct.*, vol. 32, no. 17–18, pp. 2547–2570, 1995.
[66] D. Organ, M. Fleming, T. Terry, and T. Belytschko, "Continuous meshless approximations for nonconvex bodies by diffraction and transparency," *Comput. Mech.*, vol. 18, no. 3, pp. 225–235, 1996.
[67] X. Zhuang, C. Augarde, and S. Bordas, "Accurate fracture modelling using meshless methods, the visibility criterion and level sets: Formulation and 2D modelling," *Int. J. Numer. Methods Eng.*, vol. 86, no. 2, pp. 249–268, 2011.
[68] M. Fleming, Y. A. Chu, B. Moran, and T. Belytschko, "Enriched element-free galerkin methods for crack tip fields," *Int. J. Numer. Methods Eng.*, vol. 40, no. 8, pp. 1483–1504, 1997.
[69] B. N. Rao and S. Rahman, "A coupled meshless-finite element method for fracture analysis of cracks," *Int. J. Press. Vessel. Pip.*, vol. 78, no. 9, pp. 647–657, 2001.
[70] S. Geniaut and E. Galenne, "A simple method for crack growth in mixed mode with X-FEM," *Int. J. Solids Struct.*, vol. 49, no. 15–16, pp. 2094–2106, 2012.

[71] H. Nguyen-Xuan, G. R. Liu, N. Nourbakhshnia, and L. Chen, “A novel singular ES-FEM for crack growth simulation,” *Eng. Fract. Mech.*, vol. 84, pp. 41–66, 2012.
[72] M. Pustejovsky, “Fatigue Crack Propagation in Titanium Under General In-Plane Loading - II: Analysis,” *Eng. Fract. Mech.*, vol. 11, no. 1, pp. 17–31, 1979.
[73] Z. J. Yang, X. F. Wang, D. S. Yin, and C. Zhang, “A non-matching finite element-scaled boundary finite element coupled method for linear elastic crack propagation modelling,” *Comput. Struct.*, vol. 153, pp. 126–136, 2015.